

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

О.В. Непомнящий

подпись

инициалы, фамилия

«\_\_»\_\_\_\_\_2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника

код и наименование направления

Микроконтроллерная система управления транспортным роботом.

Логистика

тема

Руководитель

\_\_\_\_\_

подпись, дата

доцент, канд.техн.наук

должность, ученая степень

В.Г. Середкин

инициалы, фамилия

Выпускник

\_\_\_\_\_

подпись, дата

Е.А. Минин

инициалы, фамилия

Нормоконтролер

\_\_\_\_\_

подпись, дата

доцент, канд.техн.наук

должность, ученая степень

В.И. Иванов

инициалы, фамилия

Красноярск 2018

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий  
институт

---

Вычислительная техника  
кафедра

---

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О.В. Непомнящий  
подпись                      инициалы, фамилия

«\_\_» \_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**в форме бакалаврской работы**

---

бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту \_\_\_\_\_ Минину Евгению Анатольевичу  
фамилия, имя, отчество

Группа \_\_\_\_\_ КИ14-07Б \_\_\_\_\_ Направление (специальность) \_\_\_\_\_ 09.03.01  
номер код

«Информатика и вычислительная техника»  
наименование

Тема выпускной квалификационной работы \_\_\_\_\_ Микроконтроллерная система  
управления транспортным роботом. Логистика

Утверждена приказом по университету № \_\_\_\_\_ от

Руководитель ВКР \_\_\_\_\_ В.Г. Середкин, к.т.н., доцент кафедры ВТ  
инициалы, фамилия, должность, учебное звание и место работы

Исходные данные для ВКР: \_\_\_\_\_ транспортный коридор с многоуровневыми  
стеллажами и гаревым покрытием пола, тип микроконтроллера –  
ATmega8535, датчики номеров стеллажей 1шт на каждый стеллаж, датчики  
расстояния от транспортного робота до стеллажей, функциональные  
клавиши, индикатор аварийных ситуаций, исполнительные устройства – два  
электродвигателя, система питания – аккумуляторные батареи

Перечень разделов для ВКР: 1 Анализ задания на ВКР; 2 Информационная  
модель системы; 3 Разработка функциональной схемы; 4 Разработка схемы  
электрической принципиальной; 5 Разработка программного обеспечения

Перечень графического материала: \_\_\_\_\_ презентация доклада выступления

Руководитель ВКР \_\_\_\_\_ В.Г. Середкин  
подпись инициалы и фамилия

Задание принял к исполнению \_\_\_\_\_ Е.А. Минин  
подпись инициалы и фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2018 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Микроконтроллерная система управления транспортным роботом. Логистика» содержит в себе 65 страницы текстового документа, 3 приложения, 41 использованных источников, 24 иллюстрации, 7 таблиц.

МИКРОКОНТРОЛЛЕР, ТРАНСПОРТНЫЙ РОБОТ, ДАТЧИК, ЭЛЕКТРОДВИГАТЕЛЬ, СТЕЛЛАЖ, СХЕМА.

Цель работы: моделирование и прототипирование микроконтроллерной системы управления транспортным роботом в складских помещениях.

Задачи:

1. Выполнить анализ существующих аналогов транспортного робота и обоснование выбора принятого решения;

2. Разработать информационную модель системы управления транспортным роботом;

2.1 Разработать структурную схему системы управления транспортным роботом;

3. Разработать функциональную схему системы управления транспортным роботом;

3.1 Выбрать компоненты для системы управления транспортным роботом;

4. Разработать схему электрическую принципиальную прототипа системы управления транспортным роботом;

5. Разработать программное обеспечение для прототипа системы управления;

5.1 Разработать блок-схему алгоритма функционирования прототипа системы управления транспортным роботом;

5.2 Моделирование прототипа в выбранной среде;

5.3 Разработка программных модулей для моделирования прототипа

В результате выполнения работы разработана схема электрическая принципиальная прототипа системы управление транспортным роботом. Промоделирована система управления транспортным роботом.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Анализ задания на выпускную квалификационную работу .....	7
1.1 Цель и задачи выпускной квалификационной работы.....	7
1.2 Обзор существующих аналогов транспортных роботов.....	7
1.2.1 Kiva .....	8
1.2.2 ОТТО .....	8
1.2.3 Буксир-1 .....	9
1.3 Обоснование выбора принятого решения .....	10
1.4 Задание на выпускную квалификационную работу .....	11
2 Информационная модель системы управления транспортным роботом .....	12
2.1 Разработка структурной схемы системы .....	13
3 Разработка функциональной схемы системы.....	15
3.1 Описание выбора компонентов для системы.....	15
3.1.1 Описание микроконтроллера .....	15
3.1.2 Обоснование выбора дисплея .....	16
3.1.3 Обоснование выбора датчика расстояния .....	18
3.1.4 Обоснование выбора функциональных кнопок.....	20
3.1.5 Обоснование выбора индикатора аварийных ситуаций .....	21
3.1.6 Обоснование выбора GSM-модуля .....	22
3.1.7 Обоснование выбора электродвигателя.....	23
3.1.8 Обоснование выбора датчиков номеров стеллажей.....	27
3.1.9 Обоснование выбора источника питания .....	29
3.2 Функциональная схема системы .....	32
4 Разработка схемы электрической принципиальной прототипа системы управления .....	34
5 Разработка программного обеспечения для системы управления.....	37
5.1 Разработка граф-схемы алгоритма работы системы и подпрограмм ....	37
5.2 Моделирование прототипа системы .....	42

ЗАКЛЮЧЕНИЕ .....	46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	47
ПРИЛОЖЕНИЕ А .....	51

## ВВЕДЕНИЕ

В современном мире технологии занимают центральное место в жизни человека. Благодаря технологиям темпы развития очень быстро растут в различных сферах: образование, культура, наука, спорт, строительство, медицина. Так же, на сегодняшний день очень актуальна тема роботов и робототехники в целом. Ещё совсем недавно эти устройства существовали лишь на страницах научно – фантастических книг. Сегодня на многих предприятиях роботы активно принимают участие во всех сферах производства, упрощая работу человека.

Робот – это автоматическое устройство, созданное по принципам распознавания, удержания и перемещения объектов во вредной, опасной или других средах, предназначенное для осуществления различного вида операций для производства, которое действует по заранее заложенной программе и получает информацию о положении и состоянии окружающего пространства посредством датчиков, робот самостоятельно осуществляет производственные и иные вспомогательные операции, удовлетворяющие производственные потребности частично или полностью заменяющие труд человека [1].

Облегчение труда людей – не единственная «заслуга» роботов. Применение роботов увеличивает производительность труда в 2-3 раза. Кроме этого применение современных автоматических манипуляторов, в полном смысле слова «думающих», «самообучающихся» машин, даёт возможность быстро перестраивать производство на выпуск новой продукции. Использование роботов открывает перспективы создания принципиально новых технологических процессов, в которых невозможно непосредственное участие человека. Например, человек при стоградусной жаре или в условиях сильной радиации работать не сможет, а робот – намного устойчивее к подобной среде работы. Роботы возьмут на себя практически весь неквалифицированный труд. Они заменят человека на



тяжёлой, опасной, монотонной, малоинтересной работе, а человек станет квалифицированным оператором, будет ими управлять, обучать их и настраивать [2].

Существует огромное количество видов роботов, самыми популярными являются следующие виды:

- Промышленные роботы предназначены для автоматизации всевозможных технологических операций (сварка, металлообработка, сборка готовых изделий и т.д.) на производстве какой-либо продукции [3];
- Строительные роботы позволяют аналогичным образом автоматизировать огромное количество различных операций, выполняемых в процессе ремонта помещений или строительства новых объектов [3];
- Бытовые роботы применяются в быту и офисах. Ярким примером бытовой автоматизированной машины является набравший приличную популярность робот-пылесос [3];
- Сельскохозяйственные роботы предназначены для выполнения трудоемких и монотонных процессов в сельском хозяйстве [3];
- Военные роботы призваны вывести вооруженные конфликты и предназначены для минимизации непосредственного участия человека в боевых действиях с целью сокращения или исключения вовсе людских потерь, а также для работы в условиях, несовместимых с возможностями человека в военных целях [3];
- Исследовательские роботы используются для сбора всевозможных видов информации об исследуемых объектах, ее переработки и передаче оператору. Объектами могут быть самыми разнообразными: поверхности планет, подводное пространство, пещеры, зараженная местность и другие труднодоступные для человека области [3];
- Перед охранными роботами ставятся задачи по защите территорий или помещений. В простейшем случае указанные роботы выполняют патрулирование охраняемых периметров и в случае фиксации проникновения злоумышленников сигнализируют об этом дежурным операторам [3];

- Транспортные роботы используются для автоматического перемещения грузов, либо автономного управления различными транспортными средствами. Транспортными роботами являются самоходные тележки, автопилоты и т.д. [3].

Из всех рассмотренных роботов самыми распространенными являются транспортные роботы. Для реализации их управления используется множество разнообразных систем, но их архитектура и техническая организация достаточно сложны.

Поэтому в выпускной квалификационной работе будет разработана микроконтроллерная система управления транспортным роботом с ограниченным функционалом, предназначенным для перевозки грузов в складских помещениях, которая будет находиться в открытом доступе.

## **1 Анализ задания на выпускную квалификационную работу**

### **1.1 Цель и задачи выпускной квалификационной работы**

Целью выпускной квалификационной работы является моделирование и прототипирование микроконтроллерной системы управления транспортным роботом в складских помещениях.

Для реализации данной цели необходимо выделить следующие задачи:

1. Анализ существующий аналогов транспортных роботов и обоснование выбора принятого решения;
2. Разработка информационной модели системы управления;
3. Разработка структурной схемы системы управления;
4. Выбор компонентов для системы управления;
5. Разработка схемы электрической принципиальной прототипа системы управления;
6. Разработка программного обеспечения для системы управления:
  - 6.1 Составление граф-схемы алгоритма работы системы;
  - 6.2 Моделирование системы управления.

В результате транспортный робот должен доставить груз к заданному программой начальной загрузки стеллажу.

### **1.2 Обзор существующих аналогов транспортных роботов**

Складские роботы – это роботы, предназначенные для ведения операций на складах, а также существуют комплексные решения автоматизации складов, использующие, например, промышленные манипуляторы, мобильные роботизированные тележки, паллетайзеры и прочую технику [4]. Каждое из этих решений имеет несколько разновидностей. Автоматизация складов идет высокими темпами, нет

сомнений, что в ближайшие годы здесь также останется мало вакансий для людей.

Были рассмотрены несколько транспортных роботов, которые предназначены для ведения операций в складских помещениях:

- Kiva;
- ОТТО;
- Буксир-1.

### 1.2.1 Kiva

Грузовой робот Kiva, компании Kiva Systems (приобретена Amazon Robotics), способен самостоятельно передвигаться по складскому помещению со скоростью порядка 8 км/ч, перевозя грузы до 317 кг. Вес устройства – около 145 кг, высота – 0.4 метра. Для использования робота Kiva требуется специальная разметка на полу помещения склада [5]. На рисунке 1 изображен внешний вид робота Kiva.



Рисунок 1 – Транспортный робот Kiva

### 1.2.2 ОТТО

Робот ОТТО, компании Clearpath Robotics, – это робототранспортер, способный перевозить грузы весом примерно до 1.5 тонн по территории складов полностью автономно. ОТТО не требует нарисованных на полу

штрихкодов или полос, как, например, складские роботы Kiva Systems, верхнего освещения, маяков на стенах. Спереди и сзади платформа оснащена лазерными локаторами, способными измерять расстояния до объектов на дистанции до 20 метров. ОТТО готов справиться с изменениями в расстановке объектов на территории, а также двигаться, не сталкиваясь с различными объектами. ОТТО снабжен мигающими светодиодами и вежливо попросит себя пропустить, если путь ему преграждают люди. Собственный вес робота – 500 кг, максимальная скорость – до 2 м/с [6]. На рисунке 2 изображен внешний вид робота ОТТО.



Рисунок 2 – Робот ОТТО

### 1.2.3 Буксир – 1

Робот компании "Стандарт-ПАК" Буксир – 1 способен перемещать грузы, оснащен датчиками движения и передвигается со скоростью от 0,5 м/с. Суммарная мощность двигателей max 80 Вт. Система питания робота 12 В. Работает по определенной или переналаживаемой программе. Старт производится кнопкой пуск или легким сдвигом робота по ходу движения. Оснащен системой обнаружения препятствий на пути следования и остановки, индикатором заряда АКБ, кнопкой включения питания, разъемом

для подключения зарядки, световой индикацией. Робот ориентируется, как по линии (нанесенной на пол или созданной системой отражателей), так и другим системам определения положения робота в пространстве на базе Arduino [7]. На рисунке 3 изображен внешний вид робота Буксир-1.



Рисунок 3 – Робот Буксир-1

### 1.3 Обоснование выбора принятого решения

Проанализировав аналоги можно сделать вывод, что большинство транспортных роботов передвигается по специальным разметкам на полу, это не совсем удобно, т.к. эти разметки необходимо рисовать по всему складскому помещению. Одним из достоинств аналогов является то, что они имеют специальный датчик, который обнаруживает препятствия на пути, и, если на пути робота встретилось препятствие – загорается светодиод. Один из роботов, может работать по переналаживаемой программе, это удобно, ведь когда-то необходимо, чтобы робот, например, обнаруживал препятствия на большем расстоянии. Но ни у одного из этих роботов нет возможности информировать работника складского помещения о наличии аварийной ситуации.

В складских помещениях необходимо доставлять очень тяжелые грузы, которые не под силу человеку. Создаваемая система будет минимальна в используемом оборудовании и тем самым будет финансово

доступна. Из этого следует, что из предоставленного множества транспортных роботов их актуальность от этого не пропадает.

#### **1.4 Задание на выпускную квалификационную работу**

Исходные данные:

- Движение транспортного робота будет осуществляться по транспортному коридору со стеллажами и гаревым покрытием пола. Гаревое покрытие обеспечивает наименьшее проскальзывание колес по ходу движения робота;
- Для обработки данных и управления всей системой будет осуществлять микроконтроллер Atmel ATmega8535;
- Определение номера необходимого стеллажа будет происходить с помощью датчиков номеров стеллажей;
- Для предотвращения столкновения транспортного робота со стеллажом необходимы датчики расстояния от транспортного робота до стеллажей;
- Обеспечение ввода нужного стеллажа, а также команды доставки груза будет обеспечиваться функциональными кнопками;
- В процессе доставки груза на пути робота могут возникнуть аварийные ситуации, о них будет сообщать индикатор аварийных ситуаций;
- Движения робота будет обеспечиваться двумя электродвигателями (по одному на левое и правое колеса);
- Источником питания будут являться аккумуляторные батареи.

## **2 Информационная модель системы управления транспортным роботом**

На основе разрабатываемой системы управления транспортным роботом можно составить информационную модель данной системы.

После загрузки груза на платформу транспортного робота оператор вводит номер необходимого стеллажа с помощью функциональных клавиш «0-9», который отображается на дисплее. Далее, оператор нажимает функциональную кнопку опправки груза, и исполнительные устройства начинают движение робота по транспортному коридору.

По ходу движения датчики расстояния от транспортного робота до стеллажей контролируют безопасное расстояние. Если расстояние слева от робота до стеллажа меньше заданного, исполнительное устройство поворачивает вправо на определенное количество градусов, как только это расстояние станет больше заданного, исполнительное устройство поворачивает влево на столько же градусов, на сколько оно повернуло вправо. Аналогично и для расстояния справа от робота.

Наличие препятствий на пути контролирует датчик расстояния спереди робота. Если это расстояние меньше или равно заданному, то происходит остановка робота и включается индикатор аварийной ситуации, вместе с этим поступает сигнал на GSM-модуль, который посылает сообщение оператору о наличии аварийной ситуации. После того, как препятствие на пути робота убрано, датчик расстояния измеряет расстояние спереди робота, если оно больше заданного, то выключается индикатор аварийной ситуации и движение робота возобновляется.

Параллельно с этим робот считывает датчики номеров стеллажей методом последовательного счёта и сравнивает с введенным номером, если эти значения совпадают, то это значит, что робот достиг заданного стеллажа и осуществил доставку груза к нему.



## 2.1 Разработка структурной схемы системы

Исходя из информационной модели пункта 2 разрабатываемой системы управления транспортным роботом можно составить структурную схему системы управления, которая изображена на рисунке 4.

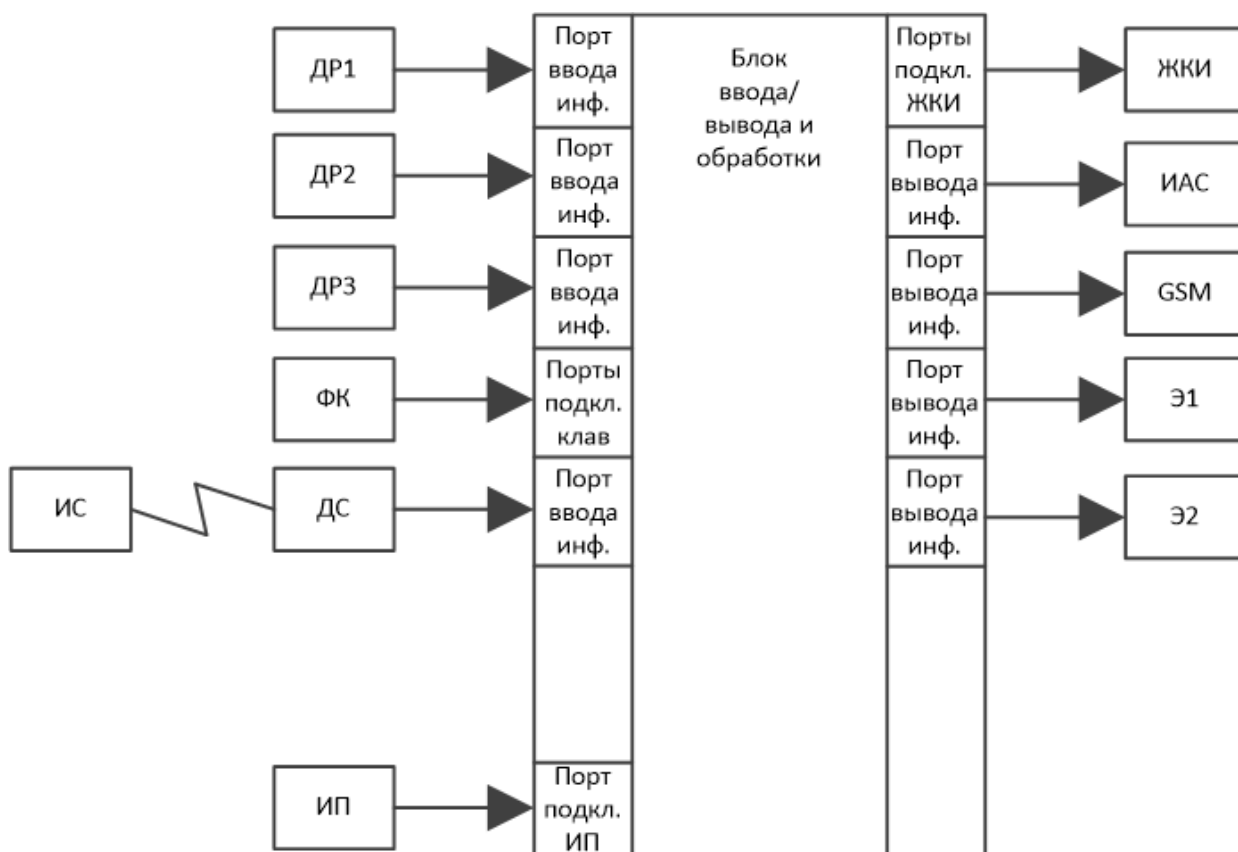


Рисунок 4 – Структурная схема системы управления

Схема, представленная на рисунке 4, включает в себя следующие обозначения:

- Блок ввода/вывода и обработки – принимает, посылает и обрабатывает необходимую информацию;
- ДР1, ДР2 – датчики расстояния слева и справа от транспортного робота до стеллажей;
- ДР3 – датчик расстояния спереди транспортного робота. Контролирует наличие препятствий на пути;

- ФК – функциональные клавиши с цифрами от 0 до 9, с кнопкой доставки груза и кнопкой сброса системы;
- ИС – идентификатор стеллажа;
- ДС – датчик стеллажа. Принимает информацию от ИС и передает её для дальнейшей обработки в блок ввода/вывода;
- ИП – источник питания;
- ЖКИ – жидкокристаллический индикатор;
- ИАС – индикатор аварийных ситуаций;
- GSM – GSM-модуль;
- Э1, Э2 – электродвигатели для левого и правого колеса.

### **3 Разработка функциональной схемы**

#### **3.1 Обоснование выбора компонентов для системы**

Исходя из структурной схемы, составленной в пункте 2.1, будет осуществляться выбор компонентов для системы управления.

##### **3.1.1 Описание микроконтроллера**

Микроконтроллер – это специальная микросхема, предназначенная для управления различными электронными устройствами. Микроконтроллеры объединяют в себя процессор, память, ПЗУ и периферию внутри одного корпуса, внешне похожего на обычную микросхему. С тех пор производство микроконтроллеров ежегодно во много раз превышает производство процессоров, а потребность в них не снижается. Микроконтроллеры выпускают десятки компаний, причем производятся не только современные 32-битные микроконтроллеры, но и 16, и даже 8-битные [8].

На сегодняшний день существует более 200 модификаций микроконтроллеров, выпускаемых двумя десятками компаний. Популярностью у разработчиков пользуются 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel [8].

По заданию необходимо использовать микроконтроллер фирмы Atmel ATmega8535, основные характеристики которого представлены ниже:

- Развитая RISC архитектура: 130 команд, большинство исполняемых за один машинный такт; 32x8 рабочих регистра; до 16 MIPS производительность;
- Энергонезависимая память программ и память данных: 8Кб FLASH памяти программ, 512б EEPROM, 512б внутренней SRAM;

- Периферия: два 8-битных таймера/счётчика; один 16-битный таймер/счетчик, программированный последовательный USART, 8 канальный, 10-битный АЦП;
- 32 программируемых входа-выхода;
- Рабочее напряжение питания 2.7В-5.5В, мощность 1Вт.

На рисунке 5 изображен внешний вид микроконтроллера ATmega8535.



Рисунок 5 – Микроконтроллер ATmega8535

### 3.1.2 Описание выбора дисплея

Для отображения номера введенного стеллажа оператором необходим дисплей. Ниже представлены характеристика некоторых дисплеев.

- Монохромный дисплей ST7567 имеет разрешение 128x64 (по 128 символов на каждые 64 строки). Для взаимодействия с этим контроллером используется последовательная шина SPI. На экране дисплея можно отображать текст, картинки и различные фигуры с помощью соответствующей библиотеки. Для питания дисплея используется напряжение 3,3В. Размер данного дисплея 60x55x20 мм. Стоимость данного дисплея около 2000р [10];
- Жидкокристаллический индикатор HD44780 – символьный ЖКИ имеет разрешение 16x2 (по 16 символов на каждые 2 строки) на базе контроллера HD44780 разработанный фирмой Hitachi. Обладает двумя

строками для вывода символов, по 40 символов на каждую строку. Имеется возможность подключения как по 4-разрядной шине данных (данные передаются за два такта), так и по 8-разрядной (данные передаются за один такт), работает от напряжения питания 5В. Средняя мощность 0,5Вт. Стоимость данного дисплея около 200р [11];

Из рассмотренных выше дисплеев лучше всего подходит LCD дисплей HD44780. Он очень дешевый по сравнению с дисплеем ST7567. Выводимой информации на экран будет не много, хоть дисплей ST7567 и позволяет выводить картинки и различные фигуры. Так же дисплей HD44780 может быть подключен по 4-разрядной шине данных, что позволяет сэкономить порты ввода/вывода микроконтроллера. Кроме этого, LCD дисплей HD44780 является самым популярным у любителей электроники.

Работа LCD дисплея HD44780 происходит следующим образом: на вход D0-D7 приходит информация, которая может быть данными, если на входе RS логическая единица, или командой, если на входе RS логический ноль. Далее обмен информацией происходит побайтно, микроконтроллер выставляет на входы D0-D7 логические единицы и нули, далее на вход E формируется стробирующий импульс, по заднему фронту которого LCD считывает данные [12]. На рисунке 6 изображён внешний вид LCD дисплея HD44780.

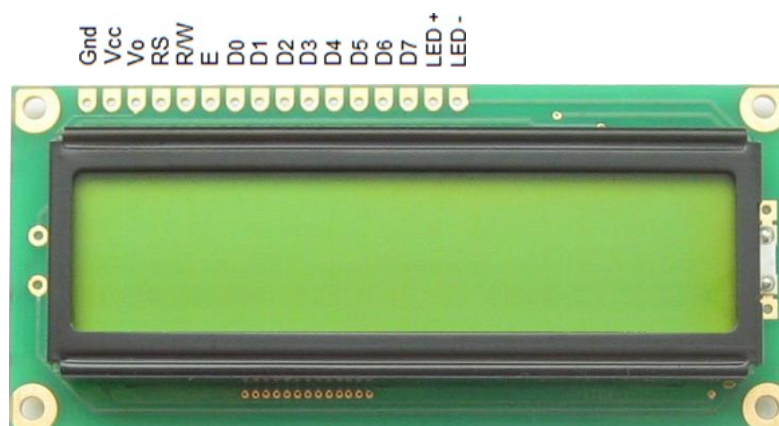


Рисунок 6 – LCD дисплей HD44780

### 3.1.3 Описание выбора датчика расстояния

На данный момент на отечественном рынке специального оборудования можно приобрести различные виды датчиков необходимых для измерения расстояния. Ниже представлены следующие разновидности датчиков:

- Оптические датчики расстояния излучают световой пучок, направленный на объект или специальный отражатель и анализируют отраженный свет. Основной принцип работы оптический датчиков расстояния основан на измерении скорости прохождения луча, который отражается от специальной отражающей метки, расположенной на объекте, или же непосредственно от самого объекта. Оптические датчики имеют большой рабочий диапазон (до 130 м) и высокое разрешение (1 мм). Основным недостатком является чувствительность к направленности датчика [13];

- Ультразвуковые измерители – идеальное решение для определения положения и удаленности объекта в тяжелых условиях эксплуатации с точностью до миллиметра. Они не подвержены воздействиям окружающей среды и позволяют проводить измерения в запыленных, задымленных помещениях, а также в помещениях с высоким уровнем шума. Более того, датчики позволяют измерять расстояние до объектов любой формы, цвета и размера, а также выполненных из различных материалов. Диапазон срабатывания датчиков очень широк: от 100 мм до 6 м [14];

- Главное достоинство магнитных датчиков расстояния – большое расстояние срабатывания при компактных габаритных размерах датчика. Единственное условие, которое накладывает применение этого типа датчика расстояния, – это оснащение объекта измерений постоянным магнитом. Между объектом с постоянным магнитом и датчиком могут быть расположены любые внешние объекты. Имеют диапазон срабатывания от 0 до 120мм [15];

Т.к. создаваемая система создается для использования транспортного робота в складских помещениях, где достаточно высокий уровень шума, пыли и прочих помех для измерения расстояния, то ультразвуковой датчик расстояния подходит для использования. Он не подвержен воздействиям окружающей среды, хорошо пригоден для определения положения и удаленности объекта в тяжелых условиях эксплуатации. В складских помещениях расстояние между стеллажа 2,2-3,5м [16], а выбранный тип датчика позволяет измерять от 100мм до 6м.

В таблице 1 представлены характеристики некоторых ультразвуковых датчиков расстояния.

Таблица 1 – Характеристики ультразвуковых датчиков расстояния

Название	Напряжения питания, В	Диапазон обнаружения, см	Мощность, Вт
HC-SR04	5	2-500	0,2
SEN-13959	5	2-400	0,15
POLOLU PARALLAX PING	4,5-6	2-300	0,15

Исходя из представленных датчиков в таблице 1 в качестве датчика безопасного расстояния от транспортного робота до стеллажей был выбран ультразвуковой датчик расстояния HC-SR04. Он обладает напряжением питания 5В и диапазоном обнаружения объекта до 5м, что подходит для расстояний между транспортным роботом и стеллажом (2,2-3,3м).

Также, для предотвращения аварийной ситуации, данный датчик расстояния будет служить контролем расстояния впереди робота.

Данный датчик расстояния работает следующим образом: на выход TRIG подается сигнал высокого уровня, длительностью 10-15 микросекунд. После этого на вывод ECHO будет подан сигнал высокого уровня, длительностью пропорциональной измеренному расстоянию. То есть микроконтроллеру достаточно измерить длительность ECHO сигнала для определения расстояния до объекта. Для того чтобы перевести измеренное

значение в сантиметры, нужно продолжительность импульса поделить на 58 [20]. На рисунке 7 изображен внешний вид ультразвукового датчика расстояния HC-SR04.

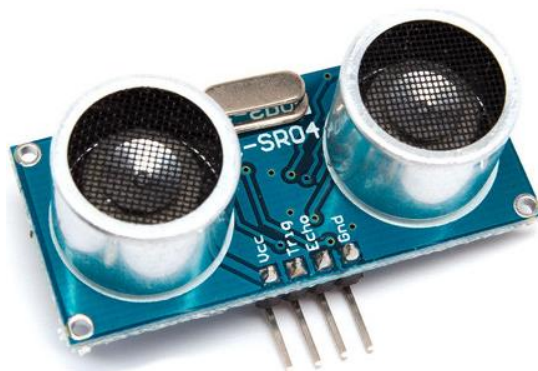


Рисунок 7 – Ультразвуковой датчик расстояния HC-SR04

### 3.1.4 Описание выбора функциональных кнопок

Функциональные кнопки необходимы для ввода номера стеллажа, отправки груза и сброса системы. Существует множество кнопок, и все они похожи друг на друга. Отличаются они лишь формой и размером. В таблице 2 представлена сравнительная характеристика некоторых кнопок.

Таблица 2 – Характеристика функциональных кнопок

Название	Рабочее напряжение, В	Рабочий ток, А	Высота, мм	Типоразмер
KLS7-TS1204-7.3-180	12	0,05	7	12x12
KLS7-TS1202-4.3-180	12	0,05	4,3	12x12
KLS7-TS6601-4.3-180	12	0,05	4,3	6x6
KLS7-TS6601-11-180	12	0,05	11	6x6

Представленные кнопки в таблице 2 имеют одинаковое рабочее напряжение и рабочий ток. Отличаются друг от друга высотой и типоразмером. Из всех представленных кнопок была выбрана кнопка KLS7-TS6601-4.3-180. Она имеет небольшие размеры, а значит будет занимать



меньше места. Таких кнопок понадобится 12 штук: 10 кнопок с цифрами от 0 до 9, 1 кнопка отправки груза и 1 кнопка сброса. На рисунке 8 изображен внешний вид кнопки.



Рисунок 8 – Функциональная кнопка

### 3.1.5 Описание выбора индикатора аварийной ситуации

В качестве индикатора аварийной ситуации может служить как дисплей, так и звуковой сигнал. Дисплей может отображать какое-либо сообщение, но это плохо информирует о наличии аварийной ситуации. Звуковой сигнал хорошо подходит для информирования о наличии аварийной ситуации на расстоянии. В таблице 3 представлены некоторые звуковые сигналы и их характеристики.

Таблица 3 – Характеристики звуковых сигналов

Название	Потребляемая мощность, Вт	Диапазон частот, Гц	Напряжение, В	Громкость, дБ
Звуковой сигнал M26	32	335	12	70
Звуковой сигнал DL50	32	310	12	75
Звуковой сигнал M100	16	350	12	73

Исходя из представленных звуковых сигналов в таблице 3 в качестве звукового индикатора аварийной ситуации подходит звуковой сигнал M100. Он обладает громкостью в 73 дБ и самой меньшей потребляемой мощностью в 16 Вт. На рисунке 9 представлен внешний вид звукового сигнала M100.



Рисунок 9 – Звуковой индикатор M100

### 3.1.6 Описание выбора GSM-модуля

GSM-модуль – это радиоприемник с установленной sim-картой любого оператора сотовой связи и контроллер для обработки поступающей и исходящей информации. В базу данных модуля заносятся телефонные номера всех пользователей, имеющие право управлять чем-либо. Современные модели поддерживают запись и хранение до 2000 номеров, более дорогие модификации способны обрабатывать до 10000 номеров [23].

Для информирования оператора о наличии аварийной ситуации подойдет любой GSM-модуль, который умеет отправлять сообщения на телефон.

Модуль SIM900 используется в различных автоматизированных системах. С помощью интерфейса UART осуществляется обмен данными с другими устройствами. Модуль обеспечивает возможность совершения звонков, обмен текстовыми сообщениями. Работа модуля реализуется на компоненте SIM900, созданным фирмой SIMCom Wireless Solution. С помощью устройства можно отслеживать маршрут транспорта совместно с ГЛОНАСС или GPS устройством. Возможность отправки смс-сообщений используется в беспроводной сигнализации и различных охранных системах [24].

Технические характеристики:

- Диапазон напряжений 4,8-5,2В;
- Поддержка 2G;
- Мощность передачи: 1Вт 1800 и 1900МГц, 2Вт 850 и 900МГц;
- В обычном режиме ток достигает 450мА, максимальный ток в импульсном режиме 2А, мощность 2,25Вт;
- GPRS multi-slot class10/8;
- Имеются встроенные протоколы TCP и UDP [24].

На рисунке 10 изображен внешний вид GSM-модуля SIM900.



Рисунок 10 – GSM-модуль SIM900

### 3.1.7 Описание выбора электродвигателя

Привод – это механизм для приведения в действие оборудования по управлению технологическими процессами с использованием электрических, пневматических или гидравлических сигналов. Это важная часть в робототехнике. Приводы, используемые в роботах, влияют на их целесообразность и производительность [28]. Рассмотрим несколько типов приводов:

- Гидравлический привод. Этот тип двигателя для приведения в движения исполнительного органа использует жидкость. Принцип работы гидропривода состоит в насосе, который создает давление рабочей жидкости в напорной магистрали, соединенной с гидродвигателем. Двигатель

преобразует давление жидкости в механическое. При этом, регуляторы управляют скоростью и направлением движения гидродвигателя. Эти приводы применяются в основном в промышленной робототехнике [28].

Преимущества данного типа привода: небольшие размеры и масса; высокая производительность – развивает силу в 25 раз выше, чем пневмопривод аналогичного размера; плавное регулирование силы; рабочая температура – от -50 до +100С;

Недостатки данного типа привода: при высоком давлении возможны утечки жидкости; высокая стоимость оборудования обслуживания; непрерывное потребление энергии; сложно отслеживать точность работы.

- Шаговый двигатель – привод, в последнее время часто используемый в робототехнике. Основное отличие между ним и всеми остальными типами двигателей состоит в способе вращения. Шаговые приводы вращаются «шагами». Каждый шаг представляет собой часть полного оборота. Эта часть зависит от механического устройства мотора и от способа управления. Использование шаговых двигателей является одним из самых простых, дешевых и легких решений для работы систем точного позиционирования. Поэтому эти двигатели очень часто используются в станках с ЧПУ и роботах [28];

Преимущества данного типа привода: точность работы, при подаче потенциалов на обмотки двигатель повернется строго на определенный угол; низкая стоимость; подходит для автоматизации отдельных механизмов и систем, где нет необходимости в высокой динамике.

Недостатки данного типа привода: присутствует проблема «проскальзывания» ротора при повышенной нагрузке на вал; ограничение шагов (максимум 1000 об/мин).

- Сервопривод. Это тип электромеханических двигателей, которые не вращаются постоянно, как шаговые, а перемещаются по сигналу в определенное положение и сохраняют его до следующего сигнала. Находят широкое применение в различных секторах робототехники – от самодельных

механизмов до сложных андроидов. В сервоприводах используется механизм обратной связи, позволяющий обрабатывать ошибки и исправлять их в позиционировании. Такая система называется следящей. Если какая-то сила оказывает давление на привод, изменяя его положение, двигатель будет применять силу в противоположном направлении, чтобы исправить возникающую ошибку. Таким образом, достигается высокая точность позиционирования [28];

Преимущества данного типа привода: более высокая точность вращения; высокая мощность; позиция механизма всегда на виду и доступна для корректирования.

Недостатки данного типа привода: сложная система подключения и управления; требует квалифицированного обслуживания; высокая стоимость.

Из представленных типов двигателей шаговый двигатель наиболее прост в использовании: при подаче потенциалов на обмотки двигатель повернется строго на определенный угол. Низкая стоимость, по сравнению с остальными двигателями, использование шаговых двигателей является одним из самых простых, дешевых и легких решений для работы систем точного позиционирования. В таблице 4 представлены характеристики некоторых шаговых двигателей.

Таблица 4 – Характеристики шаговых двигателей

Название	Выводы мотора	Шаг	Напряжение, В	Крутящий момент, кгхсм	Ток фазы, мА
42STH60-1206A	6	1,8°	7,2	не менее 6,5	1200
28STH45-0674A	4	1,8°	4,4	не менее 0,9	670
42STH47-0406A	6	1,8°	12	не менее 3,17	400
36HT20-0504MA	4	0,9°	5	не менее 0,95	500

Из представленных шаговых двигателей в таблице 4, двигатель 36HT20-0504MA имеет самый точный шаг – 0,9°, что позволяет разбить один полный оборот на 400 шагов. Биполярный двухфазный шаговый двигатель 36HT20-0504MA – привод, который способен поворачиваться на заданное

количество шагов. Выводы мотора – 4 провода: по два на каждую из обмоток, имеющих двух фаз. Рекомендованное напряжение питания мотора 5В. Мощность 5Вт. При этом ток через обмотки составит 500мА. Если устройству сложно получить указанный режим питания, то можно вращать мотор и с помощью меньшего напряжения. В этом случае соответственно снизится потребляемый ток и крутящий момент [28]. На рисунке 11 представлен внешний вид шагового двигателя 36НТ20-0504МА.



Рисунок 11 – Шаговый двигатель 36НТ20-0504МА

Для управления двигателями робота необходимо устройство, которое бы преобразовывало управляющие сигналы малой мощности в токи, достаточные для управления моторами. Такое устройство называют драйвером двигателей [30].

Микросхема L293D – это сборка из четырех Н-полумостов. Это может быть использовано для независимого управления двумя двигателями постоянного тока, с возможностью включения реверса. Также чип может быть использован для контроля одного биполярного шагового двигателя [30].

Управление осуществляется через транзисторно-транзисторную логику (TTL), а возвратные (flyback) диоды уже встроены в чип. Это означает, что для управления моторами понадобятся лишь сами моторы, L293D и микроконтроллер. L293D может выдавать до 600 мА на канал. Если этого недостаточно можно соединить несколько микросхем параллельно и тем самым поднять предельные характеристики [30].

Характеристики L293D:

- Напряжение питания управления ( $V_{cc1}$ ): 4,5 – 36 В;
- Напряжение питания нагрузки ( $V_{cc2}$ ):  $V_{cc1}$  – 36 В;
- Максимальный постоянный ток: 600 мА на канал;
- Максимальный пиковый ток (до 100 мкс): 1200 мА на канал [30].

На рисунке 12 представлен внешний вид драйвер L293D.



Рисунок 12 – Драйвер L293D

### 3.1.8 Описание выбора датчиков номеров стеллажей

Есть множество способов идентификации объектов, в данном случае – стеллажей, одним из таких способов является штрих-код. Штрих-код – это последовательность черных и белых полос, представляющая некоторую информацию в виде, удобном для считывания техническими средствами. Штрих-коды используются в торговле, складском учете, сборочном производстве и так далее. Для считывания штрих-кодов используются сканеры штрих-кодов. Сканер засвечивает штрих-код своим осветителем и считывает полученную информацию. Но дальность считывания таких приборов невелика и находится в пределах не более 1м. Существуют и дальнобойные сканеры штрих-кодов, но стоят они несколько десятков тысяч [31].

Поэтому в качестве идентификатора номера стеллажа будет служить источник инфракрасного излучения, который будет располагаться на стеллажах, а приемником будет фототранзистор, расположенный на транспортном роботе. Каждый раз, когда инфракрасное излучение будет попадать на фототранзистор, в микроконтроллер будет поступать сигнал о том, что очередной стеллаж достигнут. Таким образом, всякий раз, когда на фототранзистор будет попадать инфракрасное излучение, будет увеличиваться счетчик номера стеллажа.

Фототранзистор – это полупроводниковый прибор, который используется для управления электрическим током при помощи определенного оптического излучения [32]. Фототранзистор включает в себя базу, коллектор и эмиттер. При нерабочей базе переход коллектора фототранзистора смещается в обратном направлении, а переход эмиттера в прямом направлении. Прибор остается без активности до тех пор, пока луч света не осветит его базу. Чувствительность фототранзистора зависит от длины волны падающего излучения. Для кремниевых приборов максимум чувствительности находится в диапазоне 850-930нм – красный и ближний инфракрасный диапазоны [33].

Фототранзистор BPW85C обладает чувствительностью в диапазоне 620-980нм и углом чувствительности  $\pm 25^\circ$ . На рисунке 13 изображен внешний вид фототранзистора BPW85C [34].



Рисунок 13 – Фототранзистор BPW85C

Лазерное излучение – это электромагнитные излучения с длиной волны 0,2...1000 мкм: от 0,2 до 0,4 мкм – ультрафиолетовая область; свыше 0,4 до



0,75 мкм – видимая область; свыше 0,75 до 1 мкм – ближняя инфракрасная область; свыше 1,4 мкм – дальняя инфракрасная область. Источниками лазерного излучения являются оптические квантовые генераторы – лазеры [35].

Инфракрасный лазерный модуль DSP8503-6.510 имеет инфракрасный цвет свечения с длиной волны 850нм, а также точку в виде излучения, что очень хорошо для попадания на фототранзистор [36]. На рисунке 14 изображен внешний вид лазерного модуля DSP8503-6.510.



Рисунок 14 – Лазерный модуль DSP8503-6.510

### **3.1.9 Описание выбора источника питания**

АКБ или аккумуляторные батареи – это оборудование, которое состоит из нескольких аккумуляторов. Оно может накапливать, хранить и расходовать энергию. Благодаря обратимости химических процессов, происходящих внутри аккумулятора, такие устройства могут заряжаться и разряжаться многократно [25].

Сегодня производят около 30 типов аккумуляторов. Такое большое количество обуславливается возможностью применять в качестве электродов и электролитов различные химические элементы. Именно от материала электрода и состава электролита зависят все характеристики аккумулятора [25]. Ниже были рассмотрены особенности наиболее популярных типов аккумуляторных батарей:

- Свинцово-кислотные аккумуляторы сегодня наиболее актуальны благодаря дешевизне и универсальности. Сфера их применения обширна

ввиду большого количества разновидностей этого типа. В качестве отрицательно заряженных электродов здесь используется оксид свинца. Положительные электроды выполняются из свинца. Электролит – серная кислота. У свинцовых-кислотных батарей есть множество разновидностей имеющие разные технологии производства. Напряжение свинцово-кислотных аккумуляторов варьируются в пределах от 2 до 12 и даже до 125В [25];

- Литий-ионные аккумуляторы активно применяются в бытовой техники, электронных устройствах. Отрицательный анод – каменноугольный кокс. Положительный ион лития переносит заряд в таких батареях. Он может проникать в кристаллическую решетку других материй и образовывать с ними химическую связь. Преимуществом этого типа является высокая энергоемкость, низкий саморазряд и отсутствие нужды в обслуживании. Литий-ионные батареи также, как и их свинцовые аналоги имеют большое количество подтипов. В данном случае подтипы отличаются между собой составом катода и анода. Напряжение литий-ионных аккумуляторов варьируется в пределах от 2,4 до 3,7 В [25].

Свинцово-кислотный аккумулятор подходит для использования. Он обладает большой разновидностью подтипов аккумуляторов с различной величиной напряжения, в отличии от литий-ионных аккумуляторов, которые не смогут запитать микроконтроллер с напряжением 5В.

Примерное время работы робота будет варьироваться в пределах от 4 до 6 часов. Сложив все мощности составляющих компонентов системы и воспользовавшись формулой для нахождения емкости аккумулятора:  $C_{акб} = \frac{P_{нагр} \times T}{U_{акб} \times K \times K_{гр} \times K_{де}}$ , находим, что для 4 часов емкость аккумулятора будет примерно 15А\*ч, а при 6 часах – 25А\*ч., следовательно, нужен аккумулятор с емкостью в пределах 15-25А\*ч.

В таблице 5 представлены характеристики некоторых свинцово-кислотных аккумуляторов [26].

Таблица 5 – Характеристики свинцово-кислотных аккумуляторов

Название	Напряжение, В	Емкость, А*ч	Размер, мм	Срок службы, г
Alfa Battery FB 18-12	12	18	180x75x167	1-3
Delta GEL 12-20	12	20	181x77x167	10-12
AQQU HP12-116W	12	20	181x77x167	3-5
DELTA GX 12-24	12	24	175x166x125	10-12

Из представленных свинцово-кислотных аккумуляторов в таблице 5, аккумулятор DELTA GX 12-24 хоть достаточно габаритный, но зато имеет самую большую емкость в 24А\*ч и имеет срок службы 10-12 лет, что позволяет не беспокоиться о скорой смене данного аккумулятора. На рисунке 15 изображен внешний вид свинцово-кислотного аккумулятора DELTA GX 12-24.



Рисунок 15 – Аккумулятор LEOCH DJW12-0.8

Данный аккумулятор выдаёт напряжение 12В, а многие элементы работают от напряжения 5В, поэтому необходим понижающий преобразователь напряжения, который преобразовывает 12В в 5В.

Понижающий DC-DC преобразователь LM2596 работает от входного напряжения 4-40В, имеет возможность регулировки напряжения на выходе в диапазоне 1,3 до 30В, максимальный выходной ток 3А [27]. На рисунке 16 изображен внешний вид DC-DC преобразователя LM2596.



Рисунок 16 – Преобразователь LM2596

### 3.2 Функциональная схема системы

Исходя из выбранных компонентов в пункте 3.1, можно составить функциональную схему системы управления транспортным роботом, которая изображена на рисунке 17.

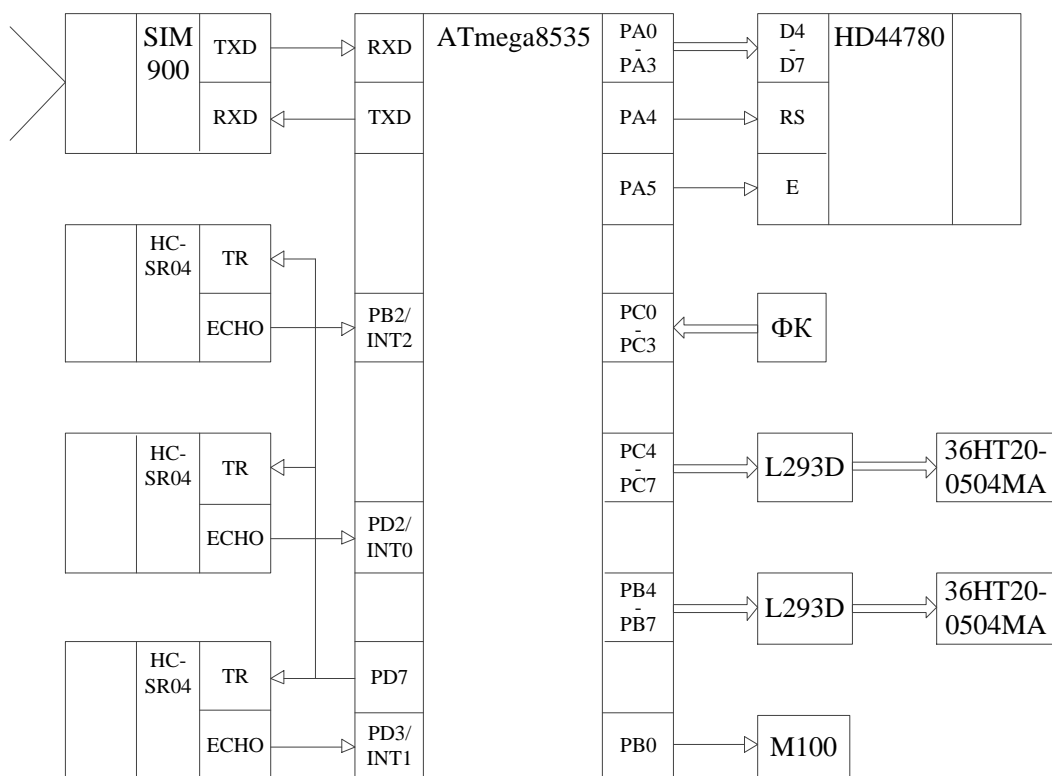


Рисунок 17 – Функциональная схема системы

Схема, представленная на рисунке 17, включает в себя следующие обозначения:

- GSM-модуль SIM900;
- Микроконтроллер ATmega8535;
- Жидкокристаллический индикатор HD44780;
- Датчики расстояния HC-SR04;
- Функциональные клавиши ФК;
- Шаговые двигатели 36HT20-0504MA;
- Драйверы шаговых двигателей L293D;
- Звуковой индикатор M100.

#### **4 Разработка схемы электрической принципиальной прототипа системы управления**

В силу того, что на этапе проектирования системы управления транспортным роботом нет возможности приобрести оборудование большей мощности, то на этапе прототипирования аккумулятор и электродвигатели будет заменены на меньшую мощность.

Система управления транспортным роботом состоит из следующих элементов:

1. Микроконтроллер ATmega8535 (DD1);
2. DC-DC преобразователь LM2596 (DD2);
3. Два драйвера шагового двигателя L293D (DD3, DD4);
4. Четыре конденсатора;
5. Кварцевый резонатор;
6. Катушка индуктивности;
7. Тринадцать резисторов;
8. Диод Шоттки;
9. Фототранзистор BPW85C;
10. Шестнадцать диодов;
11. Вилка, разъёмы.

Номиналы электронных элементов указаны в приложении «Перечень элементов».

Микроконтроллер ATmega8535 является «ядром» всей системы. К его портам подключаются светодиоды, резисторы, а также все периферийные устройства системы такие как:

1. Жидкокристаллический индикатор HD44780 – разъем SX1;
2. Звуковой индикатор M100 – разъем XS2;
3. GSM-модуль SIM900 – разъем XS3;
4. Датчики расстояния HC-SR04 – разъемы XS4, XS6, XS7;
5. Драйверы шаговых двигателей L293D – микросхемы DD3, DD4;

## 6. Кнопки – SW1-SW12.

В таблице 6 представлено подключение разъема XS1 к портам микросхемы DD1.

Таблица 6 – Подключение XS1 к DD1

Входы XS1	Порты DD1	Назначение входов XS1
D4-D7	PA0-PA3	Передача данных
E	PA5	Стrobe-эффект (1-0)
RS	PA4	RS=1 – данные, RS=0 – команда

Вход OUT разъема XS2 подключается к порту PB1 микросхемы DD1.

Входы TXD, RXD разъема XS3 подключаются к портам PD0, PD1 соответственно микросхемы DD1.

Вход ECHO разъема XS4 подключается к порту PB2 микросхемы DD1. Порт PB2 отвечает за внешнее прерывание от порта ECHO. Оно вызывается каждый раз, когда порт PB2 меняет свое логическое состояние. Вход TR подключается к порту PB0. Порт PB0 подает команду на вход TR разъема XS4.

Вход ECHO разъема XS6 подключается к порту PD2 микросхемы DD1. Порт PD2 отвечает за внешнее прерывание от порта ECHO. Оно вызывается каждый раз, когда порт PD2 меняет свое логическое состояние. Вход TR подключается к порту PB0. Порт PB0 подает команду на вход TR разъема XS6.

Вход ECHO разъема XS7 подключается к порту PD3 микросхемы DD1. Порт PD3 отвечает за внешнее прерывание от порта ECHO. Оно вызывается каждый раз, когда порт PD3 меняет свое логическое состояние. Вход TR подключается к порту PB0. Порт PB0 подает команду на вход TR разъема XS7.

Для управления шаговыми двигателями XS7, XS8 входы IN1-IN4 драйверов DD3 и DD4 подключаются к портам PD4-PD7 и PC4-PC7 соответственно микросхемы DD1.

В таблице 7 представлено подключение кнопок SW1-SW3, SW5-SW12 к портам PC0-PC3 микросхемы DD1.

Таблица 7 – Подключение кнопок к микросхеме DD1

Имя кнопки	Диод	Порт подключения	Код кнопки на портах PC3-PC0
SW5	-	PC0	0001
SW6	-	PC1	0010
SW7	VD3, VD4	PC1, PC0	0011
SW8	-	PC2	0100
SW9	VD5, VD6	PC2, PC0	0101
SW10	VD7, VD8	PC2, PC1	0110
SW11	VD9, VD10, VD11	PC2, PC1, PC0	0111
SW12	-	PC3	1000
SW1	VD12, VD13	PC3, PC0	1001
SW2	VD14, VD15	PC3, PC1	1010
SW3	VD16, VD17, VD18	PC3, PC1, PC0	1011

На DC-DC преобразователь подается +12В и на выходе напряжение составляет +5В, которое подается на все остальные разъемы. Для работоспособности к преобразователю установлены конденсаторы C1 и C2, диод Шоттки VD1 и катушка индуктивности L1. Катушка индуктивности обеспечивает необходимый номинал выходного напряжения.



## 5 Разработка программного обеспечения для системы управления

### 5.1 Разработка граф-схемы алгоритма работы системы и подпрограмм

На рисунке 18 представлена граф-схема алгоритма работы системы.

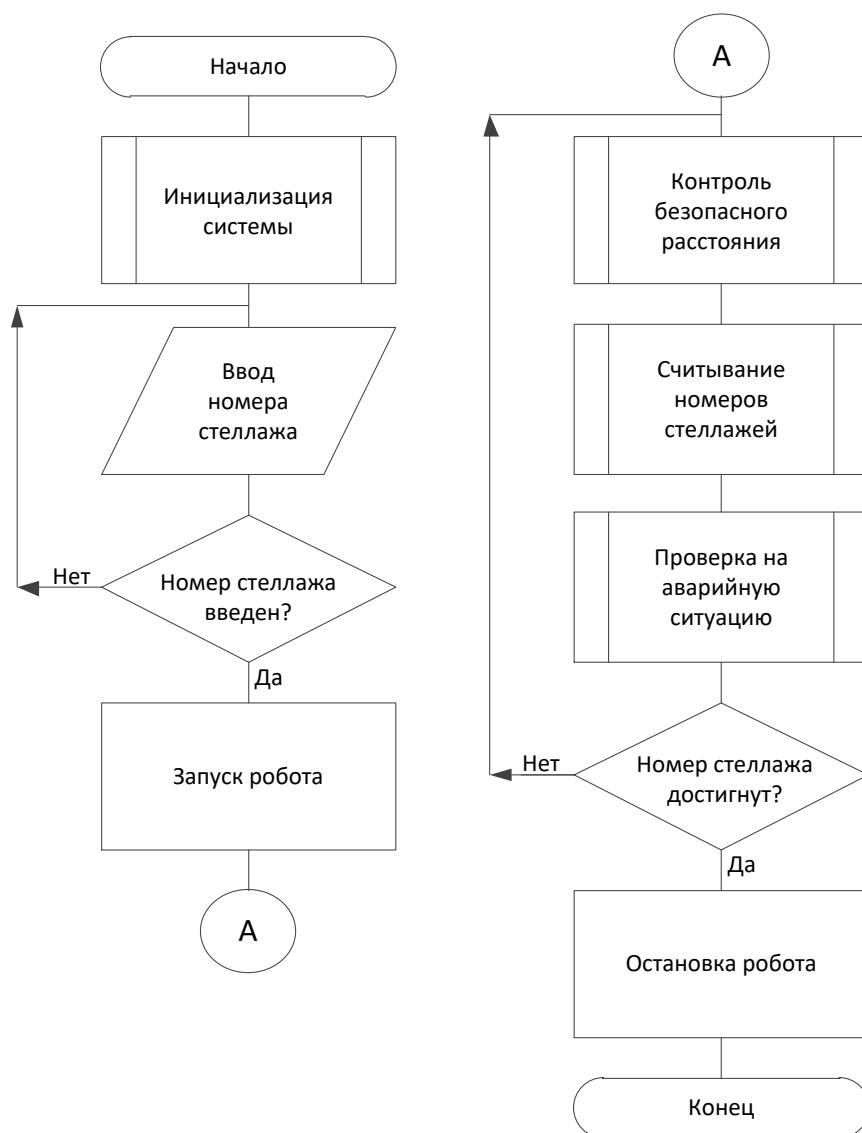


Рисунок 18 – Граф-схема алгоритма работы системы

На рисунке 19 представлена граф-схема алгоритма инициализации системы.

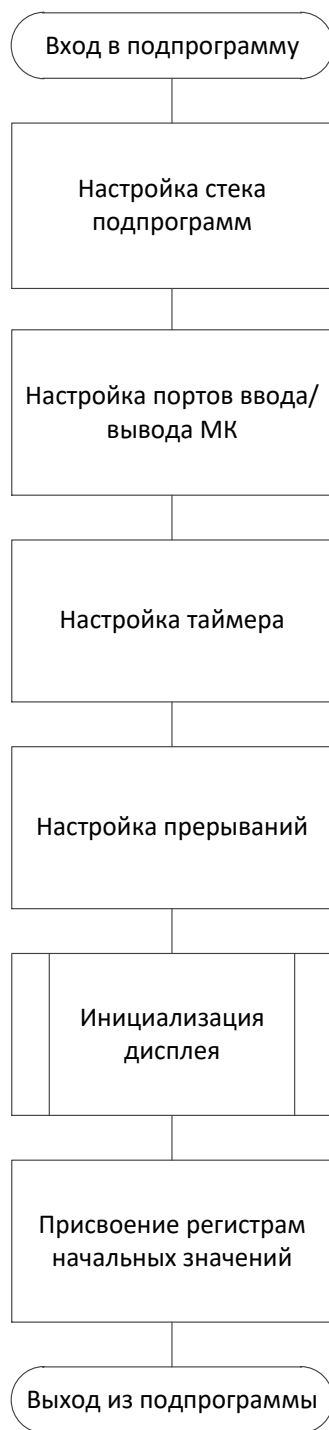


Рисунок 19 – Граф-схема инициализации системы

На рисунке 20 представлена граф-схема алгоритма инициализации дисплея.

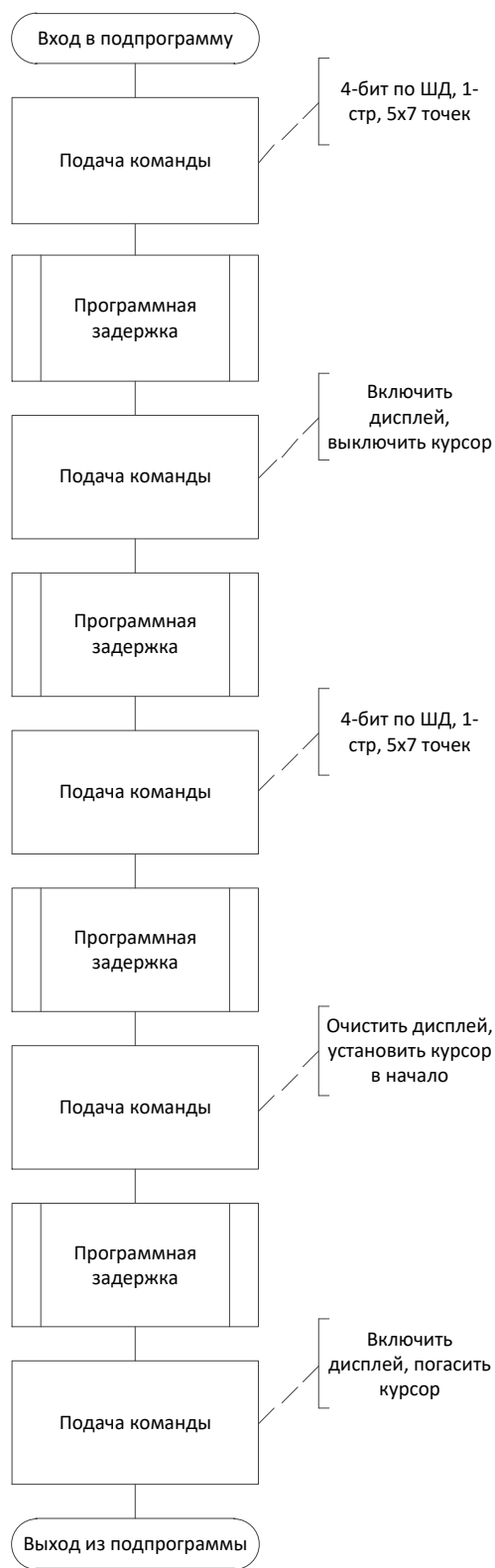


Рисунок 20 – Граф-схема инициализации дисплея

На рисунке 21 представлена граф-схема алгоритма контроля безопасного расстояния.

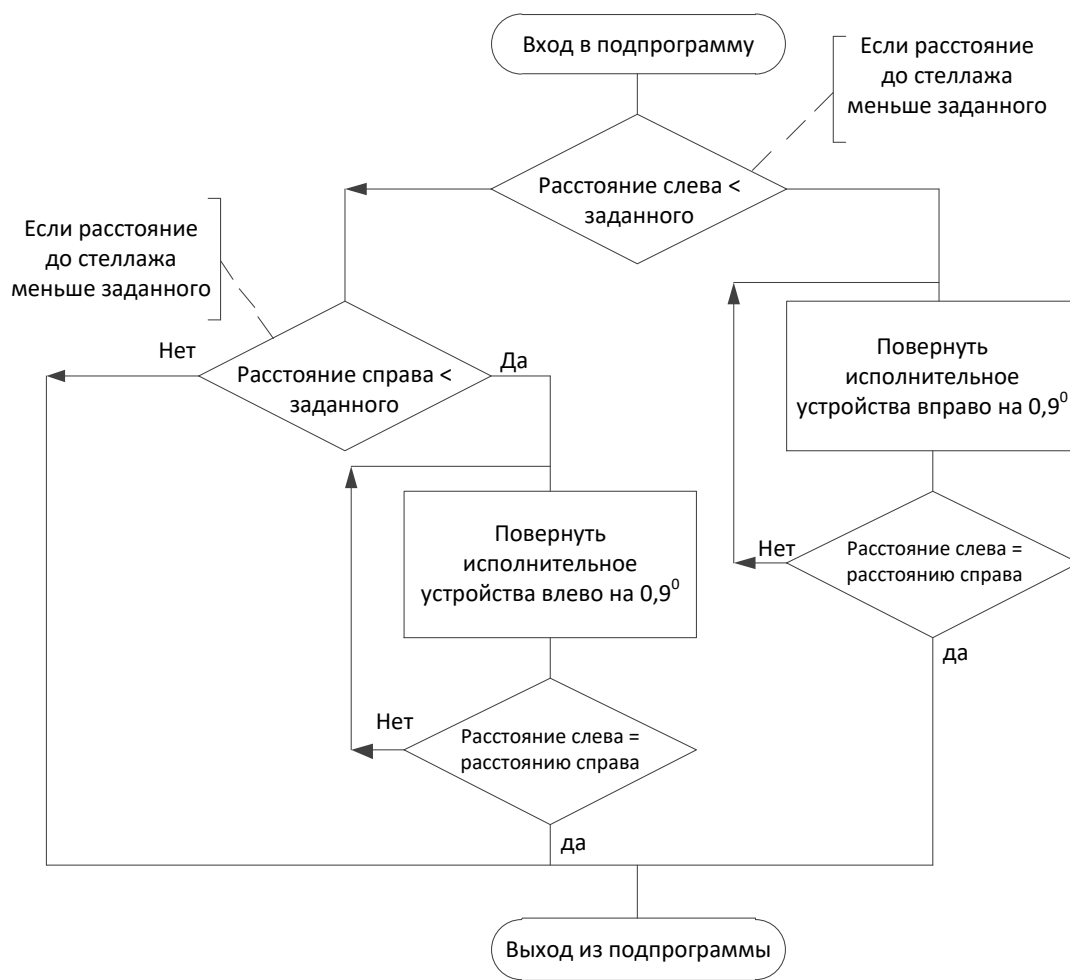


Рисунок 21 – Граф-схема контроля безопасного расстояния

На рисунке 22 представлена граф-схема алгоритма считывания номеров стеллажей.

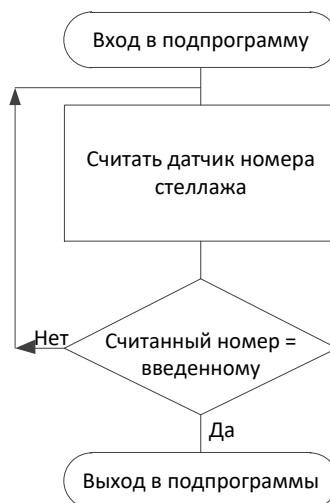


Рисунок 22 – Граф-схема считывания номера стеллажа

На рисунке 23 представлена граф-схема алгоритма проверки аварийной ситуации.

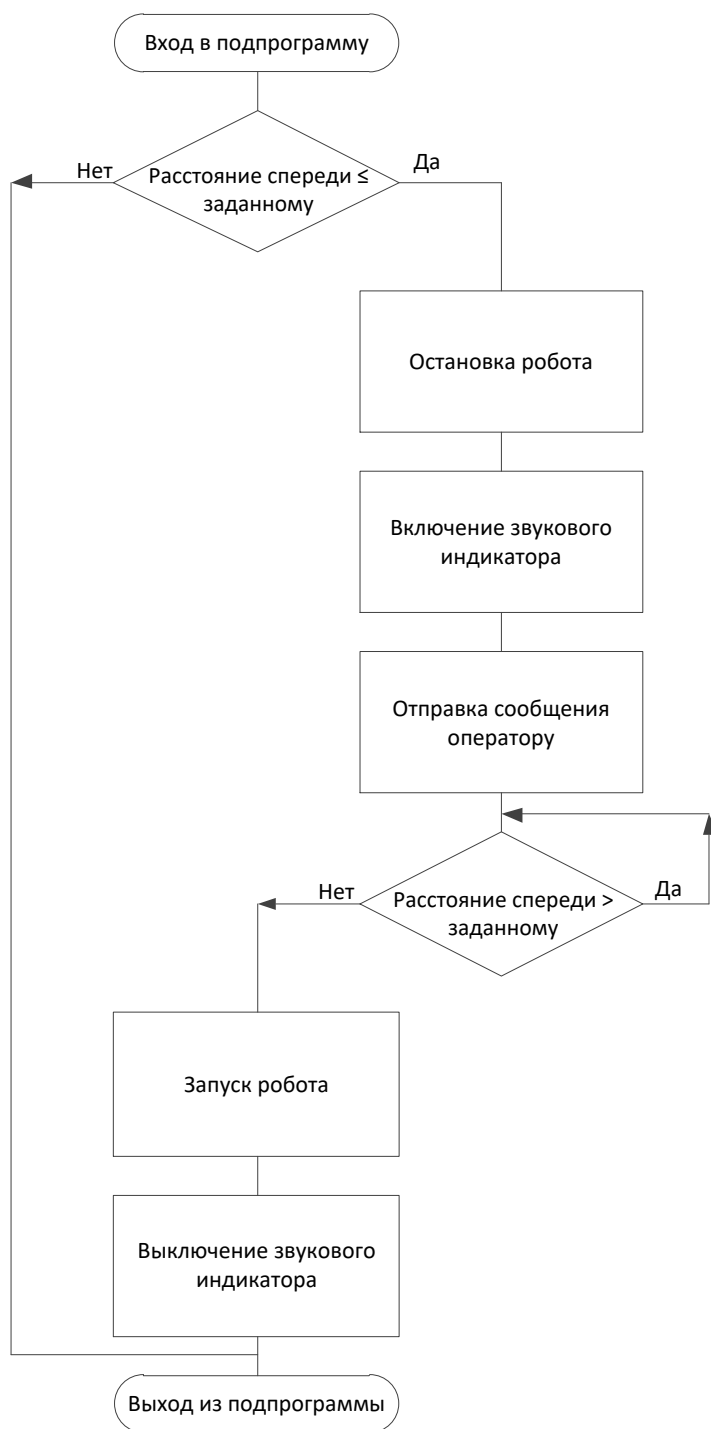


Рисунок 23 – Граф-схема проверки аварийной ситуации

## 5.2 Моделирование прототипа системы

На сегодняшний день существует множество компьютерных программ, которые предназначены для разработки различных электронных устройств. В таких программах одна из важных функций – моделирование электрических схем.

Моделирование позволяет наглядно увидеть, как создаваемая система будет работать в реальности, и как её компоненты будут взаимодействовать друг с другом. Наиболее популярными являются следующие программы: Proteus, Micro-Cap, NI Multisim, LabVIEW, LTSpice.

Proteus – мощнейшая система автоматизированного проектирования, позволяющая виртуально смоделировать работу огромного количества аналоговых и цифровых устройств. Proteus включает в себя более 6000 электронных компонентов со всеми справочными данными, а также демонстрационные ознакомительные проекты. Пакет представляет собой систему схемотехнического моделирования, базирующуюся на основе моделей электронных компонентов. Отличительной чертой пакета PROTEUS является возможность моделирования работы программируемых устройств: микроконтроллеров, микропроцессоров. Также в состав восьмой версии входит среда разработки VSM Studio, позволяющая быстро написать программу для микроконтроллера, используемого в проекте, и скомпилировать [37].

Micro-Cap – профессиональная программа аналогового, цифрового и смешанного моделирования и анализа цепей электронных устройств средней степени сложности. Интуитивно понятный интерфейс, нетребовательность к вычислительным ресурсам персонального компьютера и большой спектр возможностей послужили основой популярности Micro-Cap среди радиолюбителей, студентов и преподавателей микроэлектроники. Алгоритм работы включает в себя создание электрической цепи в графическом редакторе, задание параметров анализа и изучение полученных данных.

Графический редактор опирается на библиотеки электронных компонентов, которые можно пополнять на основе экспериментальных или справочных данных с помощью встроенного модуля Shape Editor. Моделирование включает в себя целый набор различных анализов: переходных процессов, передаточных характеристик по постоянному току, малосигнальных частотных характеристик, чувствительностей по постоянному току и многие другие [38].

NI Multisim – популярный программный пакет, позволяющий моделировать электронные схемы и разводить печатные платы. Главная особенность NI Multisim – простой наглядный интерфейс, мощные средства графического анализа результатов моделирования, наличие виртуальных измерительных приборов, копирующих реальные аналоги. Присутствуют электромеханические модели, импульсные источники питания, преобразователи мощности. В Multisim реализовано большое количество функций для профессионального проектирования микроэлектронных устройств, одной из которых является имитация работы микроконтроллеров [39].

Из рассмотренных программ для моделирования микроконтроллерной системы электрических схем была выбрана программа автоматизированного проектирования Proteus. Она обладает понятным интерфейсом, большим выбором электронных компонентов, а самое главное, данный программный продукт предоставляет встроенную среду разработки VSM Studio, позволяющую быстро написать программу для микроконтроллера, используемого в проекте, и скомпилировать, не применяя для этого стороннее программное обеспечение.

Написание кода для моделирования осуществлялось на языке Ассемблер, т.к. Proteus позволяет использовать данный язык программирования. Язык программирования Ассемблер позволяет лучше понять работу микроконтроллера и запрограммировать его так, как необходимо разработчику. Листинг программы приведен в приложении А.

На рисунке 24 представлен скриншот моделирования системы в программе автоматизированного проектирования Proteus.

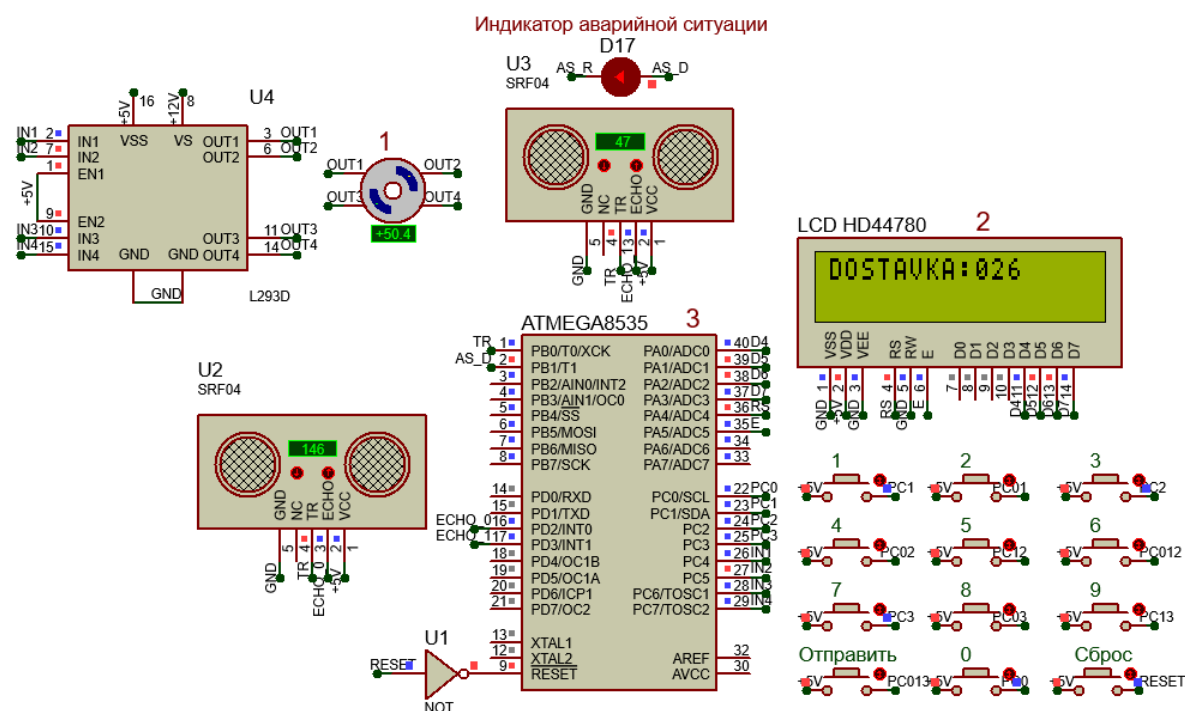


Рисунок 24 – Моделирование системы в Proteus

На рисунке 24 изображено:

- Драйвер шагового двигателя L293D (U4);
- Шаговый двигатель (1);
- Датчики расстояния SRF04 (U2, U3);
- Светодиод индикатора аварийных ситуаций (D17);
- Жидкокристаллический индикатор LCD HD44780 (2);
- Микроконтроллер ATMEGA8535 (3);
- Функциональные клавиши: «0-9», «Отправить», «Сброс»;
- Инвертор сигнала NOT (U1).

После запуска моделирования на дисплей (2) выводится сообщение «STELLAZH:». Система находится в режиме ожидания и ждёт ввода необходимого стеллажа. С помощью функциональных клавиш «0-9»,



происходит ввод номера стеллажа и его отображение на дисплее. После того, как необходимый стеллаж введён, происходит нажатие кнопки «Отправить».

Для визуальной наглядности процесса доставки груза, на дисплей выводится сообщение «DOSTAVKA:» и следом выводятся номера промежуточных стеллажей, которые робот считывает по ходу движения.

Датчик расстояния (U3) спереди робота контролирует наличие препятствий на пути. Если расстояние меньше заданного, в данном случае это 50см, то движение робота приостанавливается и загорается индикатор аварийной ситуации (D17). Как только расстояние спереди робота будет больше 50см, индикатор аварийной ситуации гаснет и движение робота возобновляется.

Контроль безопасного расстояния до стеллажей фиксируется датчиком расстояния (U2) слева от робота, чтобы предотвратить столкновение робота со стеллажом. Если это расстояние меньше заданного, в данном случае это 150см, то шаговый двигатель (1) начинается поворачивать робота вправо на определенное количество градусов, как только расстояние слева от робота станет больше 150см, то шаговый двигатель начинает поворачивать робота влево на столько же градусов, на сколько он поворачивал вправо.

После достижения необходимого стеллажа, на дисплей выводится сообщение «GRUZ DOSTAVLEN» и можно нажать кнопку «Сброс» и ввести следующий необходимый стеллаж

## **ЗАКЛЮЧЕНИЕ**

В рамках выполнения выпускной квалификационной работы проведен анализ предметной области, проанализированы существующие аналоги. Составлена информационная модель системы управления, а также на ее основе разработана структурная схема системы управления. Были выбраны компоненты из которых будет состоять система управления и на их основе разработана функциональная схема системы управления. Разработана схема электрическая принципиальная. Разработана граф-схема алгоритма работы системы и произведено моделирование системы в выбранной среде.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Словари и энциклопедии на Академике. Что такое робот [Электронный ресурс] – : Режим доступа:  
[https://dic.academic.ru/dic.nsf/enc\\_colier/6654/РОБОТ](https://dic.academic.ru/dic.nsf/enc_colier/6654/РОБОТ)
2. Xliby.ru интернет библиотека. Роботы – помощник человека [Электронный ресурс] : – Режим доступа:  
[http://www.xliby.ru/tehnicheskie\\_nauki/zanimatelnaja\\_anatomija\\_robotov/p1.php](http://www.xliby.ru/tehnicheskie_nauki/zanimatelnaja_anatomija_robotov/p1.php)
3. РобоРевю Узнайте о роботах всё! Классификация роботов [Электронный ресурс]: – Режим доступа: <http://roboreview.ru/nauka-o-robotah/klassifikatsiya-robotov.html>
4. RoboTrends. Складские роботы [Электронный ресурс] : – Режим доступа: <http://robotrends.ru/robopedia/skladskie-roboty>
5. RoboTrends. Робот Kiva [Электронный ресурс] : – Режим доступа: <http://robotrends.ru/robopedia/kiva>
6. RoboTrends. Робот ОТТО [Электронный ресурс] : – Режим доступа: <http://robotrends.ru/pub/1540/robot-otto>
7. RobotTrends. Робот Буксир–1 [Электронный ресурс] : – Режим доступа: <http://robotrends.ru/robopedia/buksir>
8. ELECTRIK INFO. [Электронный ресурс] : – Режим доступа: <http://electrik.info/main/automation/549-chto-takoe-mikrokontrollery-naznachenie-ustroystvo-princip-raboty-soft.html>
9. ATMEL 8-bit AVR Microcontroller with 8K Bytes In-System Programmable Flash. ATmega8535 ATmega8535L, Orchard Parkway, USA, 2006 г. 313 стр. // datasheet.
10. ПАКПАК. Монохромный ЖКИ 128x64. [Электронный ресурс] : – Режим доступа: <http://паспас.ru/product/dfr0287-lcd12864-shield-for-arduino/>
11. RadioProg. Символьный LCD дисплей 16x2 HD44780. [Электронный ресурс] : – Режим доступа: <http://radioprogram.ru/shop/merch/28>

12. HD44780U (LCD-II) (Dot Matrix Liquid Crystal Display Controller/Driver) HITACHI, Japan, Tokyo, 60стр. // datasheet.
13. Сенсорика. Оптические датчики расстояния. [Электронный ресурс] : – Режим доступа: <http://www.sensorica.ru/s2-1.shtml>
14. Сенсорика. Ультразвуковые датчики расстояния. [Электронный ресурс] : – Режим доступа: <http://www.sensorica.ru/s2-2a.shtml>
15. Сенсорика. Магнитные датчики расстояния. [Электронный ресурс] : –Режим доступа: <http://www.sensorica.ru/s2-6.shtml>
16. Учебный центр координального совета по логистике. [Электронный ресурс] : – Режим доступа: [http://www.ec-logistics.ru/proektirovanie\\_sklada.htm](http://www.ec-logistics.ru/proektirovanie_sklada.htm)
17. Амперка. Ультразвуковой дальномер HC-SR04 [Электронный ресурс] : – Режим доступа: <http://amperka.ru/product/hc-sr04-ultrasonic-sensor-distance-module>
18. Electronic Components. Ультразвуковой дальномер SEN-13959 [Электронный ресурс] : – Режим доступа: <https://www.tme.eu/ru/details/sf-sen-13959/moduli-datchikov/sparkfun-electronics-inc/sen-13959/>
19. Electronic Components. Ультразвуковой дальномер POLOLU PARALLAX PING. [Электронный ресурс] : – Режим доступа: [https://www.tme.eu/ru/details/parallax-ping/moduli-datchikov/pololu/parallax-ping/?brutto=1&gclid=EAIaIQobChMIq-qW9Y6H2wIVU2UZCh1e4gNmEAYYASABEgKylvD\\_BwE](https://www.tme.eu/ru/details/parallax-ping/moduli-datchikov/pololu/parallax-ping/?brutto=1&gclid=EAIaIQobChMIq-qW9Y6H2wIVU2UZCh1e4gNmEAYYASABEgKylvD_BwE)
20. ELEC Freaks, Ultrasonic Ranging Module HC – SR04 3стр. // datasheet.
21. ЧИП И ДИП. Электронные компоненты для приборов. Тактовые кнопки. [Электронный ресурс] : – Режим доступа: <https://www.chipdip.ru/catalog-show/tact-switches?x.153=jeu&x.852=pkM&sort=hits>
22. AvtoPodium. Магазин автоэлектроники. Звуковые сигналы [Электронный ресурс] : – Режим доступа: <https://avtopodium.net/catalog/stroboskopy-signal-y-sgu/signal-y-avtomobilnye>

23. ARZE.ru GSM модуль для дистанционного управления [Электронный ресурс] : – Режим доступа: <http://arze.ru/security/barriers/gsm-moduli-dlya-distantsionnogo-upravleniya.html>

24. ArduinoMaster. Все об ардуино. GSM и GPRS модули для Arduino [Электронный ресурс] : – Режим доступа: <https://arduino-master.ru/datchiki-arduino/gsm-gprs-modul-arduino/>

25. ЭЛЕКТРОСАМ.РУ Электрика и электрооборудование, электротехника и электроника – информация! Аккумуляторные батареи. Виды, особенности, устройство, применение [Электронный ресурс] : – Режим доступа:

<https://electrosam.ru/glavnaja/jelektrooborudovanie/jelektropitanie/akkumulyatornye-batarei/>

26. 1000VA Интернет-магазин Аккумуляторов и ИБП на рынке с 2004г. Аккумуляторы, аккумуляторные батареи [Электронный ресурс] : – Режим доступа: [http://www.1000va.ru/shop/akkumulyatornye\\_batarei/](http://www.1000va.ru/shop/akkumulyatornye_batarei/)

27. Texas Instruments LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator, Dallas, Texas, November 1999-Revised May 2018, 45стр. // datasheet.

28. NanoJam. Магазин роботов. 7 популярных приводов для роботов. [Электронный ресурс] : – Режим Доступа: [https://nanojam.ru/news/7\\_populyarnih\\_privodov\\_dlya\\_robotov](https://nanojam.ru/news/7_populyarnih_privodov_dlya_robotov)

29. Амперка. Шаговые двигатели. [Электронный ресурс] : – Режим доступа: <http://amperka.ru/collection/stepper-motor>

30. Мой робот. Роботы, робототехника, микроконтроллеры. Драйвер двигателей L293D [Электронный ресурс] : – Режим доступа: [https://myrobot.ru/stepbystep/el\\_driver.php](https://myrobot.ru/stepbystep/el_driver.php)

31. Добро ЕСТЬ! Справочная информация. Штриховой код (штрихкод). Описание, виды и расшифровка штрихкода. [Электронный ресурс] : – Режим доступа: <http://spravka.dobro-est.com/spravochnik/kodyi/shtrihovoy-kod-shtrihkod-opisanie-vidyi-i-rasshifrovka-shtrihkoda.html>

32. Asutpp.ru Автоматизация и электрика. [Электронный ресурс] : – Режим доступа: <https://www.asutpp.ru/avtomatizaciya-proizvodstva/fototranzistor.html>

33. ЭЛЕКТРОСАМ.РУ Электрика и электрооборудование, электротехника и электроника – информация! Фототранзисторы. Устройство. Работа. Применение. Особенности. [Электронный ресурс] : – Режим доступа: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/fototranzistor/>

34. ЧИП И ДИП. Электронные компоненты для приборов. Фототранзисторы. [Электронный ресурс] : – Режим доступа: <https://www.chipdip.ru/catalog-show/phototransistors>

35. Охрана труда. Информационный ресурс. Лазерное излучение и защита от него на производстве. [Электронный ресурс] : – Режим доступа: [http://ohrana-bgd.ru/bgdproiz/bgdproiz1\\_55.html](http://ohrana-bgd.ru/bgdproiz/bgdproiz1_55.html)

36. ЧИП И ДИП. Электронные компоненты для приборов. Лазерные модули. [Электронный ресурс] : – Режим доступа: <https://www.chipdip.ru/catalog-show/laser-modules>

37. Proteus. Официальный сайт Proteus [Электронный ресурс] : – Режим доступа: <https://www.labcenter.com/>

38. Micro-Cap. Официальный сайт Micro-Cap [Электронный ресурс] : – Режим доступа: <https://microcap.com/>

39. National Instruments. Официальный сайт NI Multisim [Электронный ресурс] : – Режим доступа: <http://www.ni.com/multisim/>

40. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 30.12.2013. – Красноярск: ИПК СФУ, 2014. - 60с.

41. Электронные курсы СФУ [Электронный ресурс] : – Режим доступа: <https://e.sfu-kras.ru/>

## ПРИЛОЖЕНИЕ А

### Листинг программы

```
;ПРИСВОЕНИЕ РЕГИСТРАМ СИМВОЛЬНЫХ ИМЕН
.include "m8535def.inc" ;Файл МК
.def Stellazh = r16      ;Рег. введенного номера стеллажа
.def Stellazh_buff = r17 ;Рег. промежуточного номера стеллажа
.def Temp = r18          ;Вспомогательный рег. 1
.def Temp_2 = r19        ;Вспомогательный рег. 2
.def Click = r20         ;Рег. контроля нажатия клавиш
.def Distance_front = r21 ;Рег. датчика расстояния спереди робота
.def Distance_left = r22  ;Рег. датчика расстояния слева от робота
.def Count_sensor = r23   ;Рег. подсчета срабатывания датчиков расстояния
.def Count_turn = r24     ;Рег. поворота

;СЕКМЕНТ КОДА
.cseg
.org $000 rjmp Start      ;Начало программы
.org $001 rjmp P_INT0     ;Прерывание INT0 (для датчика слева от робота)
.org $002 rjmp P_INT1     ;Прерывание INT1 (для датчика спереди робота)
.org $006 rjmp Timer_16   ;Вектор обработки таймера 16БИТ по сравнению
.org $013 rjmp Timer_8    ;Вектор обработки таймера 8БИТ по сравнению

;МАКРОС ВЫВОДА ИНФОРМАЦИИ НА ЖКИ
.macro Input_JKI
;Идет подсчет сотен, десятков, единиц и их вывод
clr Temp      ;Обнулили рег. подсчета
mov Temp_2, @0 ;Переместили номер стеллажа в вспомогательный рег.
Count_sot: ;Подсчет сотен
cpi Temp_2, 100 ;Сравниваем с 100
brlo Output_sot ;Если меньше - выводим кол-во сотен (метка Output_sot)
inc Temp      ;Увеличили счетчик сотен
subi Temp_2, 100 ;Вычитаем 100
rjmp Count_sot
Output_sot: ;Вывод сотен
ldi r25, 48 ;Загрузили код нуля
add r25, Temp ;Сложили со счетчиком сотен
rcall Write_dat ;Вывели на ЖКИ
clr Temp      ;Обнулили рег. подсчета
Count_des: ;Подсчет десятков
cpi Temp_2, 10 ;Сравниваем с 10
brlo Output_des ;Если меньше - выводим кол-во десятков (метка Output_des)
inc Temp      ;Увеличили счетчик десятков
```

```

    subi Temp_2, 10 ;Вычитаем 10
    rjmp Count_des
Output_des: ;Вывод десятков
    ldi r25, 48 ;Загрузили код нуля
    add r25, Temp ;Сложили со счетчиком десятков
    rcall Write_dat ;Вывели на ЖКИ

    ;Вывод единиц
    ldi r25, 48 ;Загрузили код нуля
    add r25, Temp_2 ;Сложили со счетчиком единиц
    rcall Write_dat ;Вывели на ЖКИ
.endmacro ;Конец макроса

;МАКРОС ЗАДЕРЖКИ
.macro DELAY
    ldi r27, @0
    ldi r26, @0
sd_10:
    sd_255:
        dec r26
        brne sd_255
    dec r27
    brne sd_10
.endmacro ;Конец макроса

;МАКРОС НАЖАТИЯ КЛАВИШ ;@0 - номер клавиши (49, 50...) ;@1 - First_Click (100, 200) ;@2
- Second_Click (10, 20...) ;@3 - Thirty_Click (1, 2, 3...)
.macro PRESS_KEY
    ldi r25, @0 ;Загрузили код нажатой клавиши
    rcall Write_dat ;Вывели на экран
    rcall Long_delay ;Задержка

    cpi Click, 1
    breq First_click ;Переходим на метку First_click, если не было сделано первого нажатия
    cpi Click, 2
    breq Second_click ;Переходим на метку Second_click, если не было сделано второго
нажатия
    ldi Temp, @3 ;Третье нажатие
    add Stellazh, Temp
    ldi Click, 0 ;Все три клавиши нажаты
    rjmp label_1
First_click: ;Первое нажатие
    ldi Stellazh, @1
    ldi Click, 2 ;Ожидание нажатие второй клавиши
    rjmp label_1
Second_click: ;Второе нажатие

```



```

    ldi Temp, @2
    add Stellazh, Temp
    ldi Click, 3 ;Ожидание нажатие третьей клавиши
label_1:
.endmacro ;Конец макроса

;НАЧАЛО ПРОГРАММЫ
Start:
;НАСТРОЙКА СТЕКА
ldi Temp, LOW(RAMEND)
out spl, Temp
ldi Temp, HIGH(RAMEND)
out sph, Temp

;НАСТРОЙКА ПОРТОВ ВВОДА/ВЫВОДА МК
ldi Temp, 0b11111111 ;Настройка порта А на вывод (ЖКИ)
out DDRA, Temp
out DDRB, Temp ;Настройка порта В на вывод (светодиод и выход TR для датчика
расстояния)
ldi Temp, 0b11110000 ;Настройка порта С на вывод (Двигатель) и на ввод (кнопки)
out DDRC, Temp

;НАСТРОЙКА ТАЙМЕРА 16-БИТ и 8-БИТ
ldi Temp, 0b00010010 ;Настройка 16-битного и 8-битного таймера по сравнению
out TIMSK, Temp

;Заносим в рег. сравнения число 1048 (для таймера 16-БИТ)
ldi Temp, 0b000000100 ;0b000000100
out OCR1AH, Temp
ldi Temp, 0b00011000 ;0b00011000
out OCR1AL, Temp

;Счет таймера идет с нуля (для таймера 16-БИТ)
rcall Clear_Timer_16

;Счет таймера идет с нуля (для таймера 8-БИТ)
rcall Clear_Timer_8

;НАСТРОЙКА ПЕРЕРЫВАНИЙ
ldi Temp, 0b000000101 ;Любое изменение INTO и INT1
out MCUCR, Temp

;ПРИСВОЕНИЕ РЕГИСТРАМ НАЧАЛЬНЫХ ЗНАЧЕНИЙ
ldi Click, 1 ;Ожидание нажатия первой клавиши
ldi Stellazh, 0 ;Изначально номер стеллажа равен 0
ldi Stellazh_buff, 0 ;Изначально номер промежуточного стеллажа равен 0

```

ldi Count\_sensor, 0 ;Обнулили рег. подсчета срабатывания датчиков расстояния  
ldi Count\_turn, 0 ;Обнулили рег. поворота

rcall Init\_lcd ;Инициализация дисплея  
rcall Stellazh\_info ;Вывод сообщения 'Stellazh:'

sei ;Разрешить прерывания

;БЕСКОНЕЧНЫЙ ЦИКЛ

Main:

in Temp\_2, PINC ;Считали данные из порта C  
cpi Click, 0 ;Сравнили с 0  
breq Stop\_Click ;Если все три нажатия сделаны - запрещаем нажимать клавиши ввода  
стеллажа  
cpi Temp\_2, 0b00000001 ;Если в порту C данный код - вызываем подпрограмму  
нажатия клавиши 0  
brne label\_1  
rcall Zero ;Подпрограмма нажатия клавиши 0  
label\_1:  
cpi Temp\_2, 0b00000010 ;Если в порту C данный код - вызываем подпрограмму  
нажатия клавиши 1  
brne label\_2  
rcall One ;Подпрограмма нажатия клавиши 1  
label\_2:  
cpi Temp\_2, 0b00000011 ;Если в порту C данный код - вызываем подпрограмму  
нажатия клавиши 2  
brne label\_3  
rcall Two ;Подпрограмма нажатия клавиши 2  
label\_3:  
cpi Click, 1  
breq Stop\_Click ;Если первое нажатие не произошло - запрещаем нажимать клавиши 3-  
9  
cpi Temp\_2, 0b00000100 ;Если в порту C данный код - вызываем подпрограмму  
нажатия клавиши 3  
brne label\_4  
rcall Three ;Подпрограмма нажатия клавиши 3  
label\_4:  
cpi Temp\_2, 0b00000101 ;Если в порту C данный код - вызываем подпрограмму  
нажатия клавиши 4  
brne label\_5  
rcall Four ;Подпрограмма нажатия клавиши 4  
label\_5:  
cpi Temp\_2, 0b00000110 ;Если в порту C данный код - вызываем подпрограмму  
нажатия клавиши 5  
brne label\_6  
rcall Five ;Подпрограмма нажатия клавиши 5

```

label_6:
    cpi Stellazh, 200
    brsh Click_2 ;Если число больше или равно 200 - проверяем, было ли сделано
второе нажатие (метка Click_2)
    rjmp p_2
Click_2:
    cpi Click, 2
    breq Stop_Click ;Если второе нажатие не произошло - запрещаем нажимать
клавиши 6-9
    p_2:
        cpi Stellazh, 250
        brsh Stop_Click ;Если число больше или равно 250 - то запрещаем нажимать
клавиши 6-9
            cpi Temp_2, 0b00000111 ;Если в порту С данный код - вызываем
подпрограмму нажатия клавиши 6
            brne label_7
            rcall Six ;Подпрограмма нажатия клавиши 6
            label_7:
                cpi Temp_2, 0b00001000;Если в порту С данный код - вызываем
подпрограмму нажатия клавиши 7
                brne label_8
                rcall Seven ;Подпрограмма нажатия клавиши 7
                label_8:
                    cpi Temp_2, 0b00001001 ;Если в порту С данный код - вызываем
подпрограмму нажатия клавиши 8
                    brne label_9
                    rcall Eight ;Подпрограмма нажатия клавиши 8
                    label_9:
                        cpi Temp_2, 0b00001010 ;Если в порту С данный код - вызываем
подпрограмму нажатия клавиши 9
                        brne label_10
                        rcall Nine ;Подпрограмма нажатия клавиши 9
                        label_10:
Stop_Click:
    cpi Click, 0
    brne No_Zapusk ;Переходим на метку No_Zapusk, если все три нажатия не произведены
    cpi Temp_2, 0b00001011
    brne No_Zapusk
    rcall Zapusk ;Подпрограмма запуска
    No_Zapusk:
rjmp Main

;Подпрограмма запуска
Zapusk:
    Idi Temp, 0b11000000 ;Разрешаем прерывание INT0 и INT1
    out GICR, Temp

```

```

;На вход TR подаем сигнал высокого уровня
sbi PORTB, 0
DELAY 100
cbi PORTB, 0

rcall Start_Timer_16 ;Запускаем таймер 16БИТ
rcall Start_Timer_8 ;Запускаем таймер 8БИТ
ret ;Возврат из подпрограммы

;Прерывание для датчика расстояния слева от робота
P_INT0:
sbis PIND, 2 ;Пропустить, если 2 бит в порту D установлен
rjmp label_output_0 ;Переходим на метку label_output_0, если бит в порту D очищен
rjmp label_clear_timer_0 ;Переходим на метку label_clear_timer_0, если бит в порту D
установлен
label_output_0:
inc Count_sensor ;Увеличили рег. количества срабатываний датчиков расстояния
cpi Count_sensor, 2 ;Если все датчики расстояния сработали - не считываем их значения
breq qwerty
in Distance_left, TCNT0 ;Зафиксировали расстояние слева от робота
qwerty:

;rcall Clear_Display
;Input_JKI Distance_left

cpi Distance_left, 155 ;Если расстояние меньше 150см - поворачиваем вправо
brlo turn_left ;Если больше 150см - поворачивает влево (метка label_60)
rcall Right ;Вызов подпрограммы поворота робота вправо
inc Count_turn ;Увеличили рег. поворота
rjmp label_61
turn_left:
cpi Count_turn, 0 ;Сравнили с 0
breq no_turn_left ;Переходим на метку no_turn_left, если Count_turn равно 0
rcall Left ;Вызов подпрограммы поворота робота влево
dec Count_turn ;Уменьшили рег. поворота
no_turn_left:
label_61:

cpi Count_sensor, 2 ;Если все датчики расстояния сработали - переходим на метку
label_TR_0
breq label_TR_0
rjmp label_12
label_TR_0:
clr Count_sensor ;Обнулили рег. количества срабатываний датчиков расстояния
DELAY 100

```

```

;На вход TR подаем сигнал высокого уровня
sbi PORTB, 0
DELAY 45
cbi PORTB, 0
rjmp label_12
label_clear_timer_0:
inc Count_sensor ;Увеличили рег. количества срабатываний датчиков расстояния
cpi Count_sensor, 2 ;Если все датчики расстояния сработали - переходим на метку
label_null_0
breq label_null_0
rjmp label_12
label_null_0:
;Счет таймера идет с нуля
rcall Clear_Timer_8
clr Count_sensor ;Обнулили рег. количества срабатываний датчиков расстояния
label_12:

reti ;Возврат из прерывания

;Прерывание для датчика расстояния спереди робота
P_INT1:
sbis PIND, 3 ;Пропустить, если 2 бит в порту D установлен
rjmp label_output_1 ;Переходим на метку label_output_1, если бит в порту D очищен
rjmp label_clear_timer_1 ;Переходим на метку label_clear_timer_1, если бит в порту D
установлен
label_output_1:
inc Count_sensor ;Увеличили рег. количества срабатываний датчиков расстояния
cpi Count_sensor, 2 ;Если все датчики расстояния сработали - не считываем их значения
breq qwerty_1
in Distance_front, TCNT0 ;Зафиксировали расстояние спереди робота
qwerty_1:

;rcall Clear_Display
;Input_JKI Distance_front

cpi Distance_front, 180 ;Если расстояние больше 50см - переходим на метку label_63
brpl label_63
sbi PORTB, 1 ;Зажигаем светодиод индикатора аварийных ситуаций
rcall Stop_Timer_16 ;Остановка таймера 16БИТ
rjmp label_62
label_63:
cbi PORTB, 1 ;Гасим светодиод индикатора аварийных ситуаций
rcall Start_Timer_16 ;Запуск таймера 16БИТ
label_62:

```

```

    cpi Count_sensor, 2 ;Если все датчики расстояния сработали - переходим на метку
label_TR_1
    breq label_TR_1
    rjmp label_14
label_TR_1:
    clr Count_sensor ;Обнулили рег. количества срабатываний датчиков расстояния
    DELAY 100
    ;На вход TR подаем сигнал высокого уровня
    sbi PORTB, 0
    DELAY 45
    cbi PORTB, 0
    rjmp label_14
label_clear_timer_1:
    inc Count_sensor ;Увеличили рег. количества срабатываний датчиков расстояния
    cpi Count_sensor, 2 ;Если все датчики расстояния сработали - переходим на метку
label_null_1
    breq label_null_1
    rjmp label_14
label_null_1:
    ;Счет таймера идет с нуля
    rcall Clear_Timer_8
    clr Count_sensor ;Обнулили рег. количества срабатываний датчиков расстояния
label_14:
reti ;Возврат из прерывания

;Прерывание таймера 8БИТ по сравнению
Timer_8:
reti ;Возврат из прерывания

;Прерывание таймера 16БИТ по сравнению
Timer_16:
    rcall Clear_Display ;Очистка дисплея
    rcall Dostavka_info ;Вывод сообщения 'DOSTAVKA:'

    cp Stellazh, Stellazh_buff ;Сравнили введенный стеллаж с текущим (увеличиваем номер
текущего стеллажа, если не равны)
    breq Dostavleno ;Переходим на метку Dostavleno, если введенный и текущий стеллаж
равны
    inc Stellazh_buff ;Увеличили номер текущего стеллажа
    Input_JKI Stellazh_buff ;Вывели текущий стеллаж на дисплей
    rjmp Ne_Dostavleno ;Переходим на метку Ne_Dostavleno
Dostavleno:
    ldi Temp, 0b00000000 ;Запрещаем прерывание INT0 и INT1
    out GICR, Temp
    rcall Stop_Timer_8 ;Остановка таймера 8БИТ
    rcall Stop_Timer_16 ;Остановка таймера 16БИТ

```

```

    rcall Clear_Display ;Очистка дисплея
    rcall Gruz_info     ;Вывод сообщения 'GRUZ DOSTAVLEN'
Ne_Dostavleno:
    ;Счет таймера идет с нуля
    rcall Clear_Timer_16
reti ;Возврат из прерывания

```

```

;Подпрограмма поворота вправо
Right:

```

```

    ldi Temp, 0b00010000
    out PORTC, Temp
    DELAY 255
    ldi Temp, 0b00100000
    out PORTC, Temp
    DELAY 255
    ldi Temp, 0b01000000
    out PORTC, Temp
    DELAY 255
    ldi Temp, 0b10000000
    out PORTC, Temp
    DELAY 255
ret ;Возврат из подпрограммы

```

```

;Подпрограмма поворота влево
Left:

```

```

    ldi Temp, 0b10000000
    out PORTC, Temp
    DELAY 255
    ldi Temp, 0b01000000
    out PORTC, Temp
    DELAY 255
    ldi Temp, 0b00100000
    out PORTC, Temp
    DELAY 255
    ldi Temp, 0b00010000
    out PORTC, Temp
    DELAY 255
ret ;Возврат из подпрограммы

```

```

;Подпрограмма очистки дисплея

```

```

Clear_Display:
    ldi r25, 0b00000001
    rcall Write_com
ret ;Возврат из подпрограммы

```

```

;Подпрограмма запуска таймера 8БИТ

```

Start\_Timer\_8:

ldi Temp, 0b00000011 ;Запуск таймера с частотой МК/512

out TCCR0, Temp

ret ;Возврат из подпрограммы

;Подпрограмма остановки таймера 8БИТ

Stop\_Timer\_8:

ldi Temp, 0b00000000 ;Остановка таймера

out TCCR0, Temp

ret ;Возврат из подпрограммы

;Подпрограмма сброса таймера 8БИТ

Clear\_Timer\_8:

ldi Temp, 0b00000000

out TCNT0, Temp

ret ;Возврат из подпрограммы

;Подпрограмма запуска таймера 16БИТ

Start\_Timer\_16:

ldi Temp, 0b00000101 ;Запуск таймера с частотой МК/1024

out TCCR1B, Temp

ret ;Возврат из подпрограммы

;Подпрограмма остановки таймера 16БИТ

Stop\_Timer\_16:

ldi Temp, 0b00000000 ;Остановка таймера

out TCCR1B, Temp

ret ;Возврат из подпрограммы

;Подпрограмма сброса таймера 16БИТ

Clear\_Timer\_16:

ldi Temp, 0b00000000

out TCNT1H, Temp ;Старший рег.

out TCNT1L, Temp ;Младший рег.

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 0

Zero:

PRESS\_KEY 48, 0, 0, 0 ;Передаём в макрос код цифры нуля, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 1

One:

PRESS\_KEY 49, 100, 10, 1 ;Передаём в макрос код цифры один, кол-во сотен, десятков и единиц



ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 2

Two:

PRESS\_KEY 50, 200, 20, 2 ;Передаём в макрос код цифры два, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 3

Three:

PRESS\_KEY 51, 0, 30, 3 ;Передаём в макрос код цифры три, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 4

Four:

PRESS\_KEY 52, 0, 40, 4 ;Передаём в макрос код цифры четыре, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 5

Five:

PRESS\_KEY 53, 0, 50, 5 ;Передаём в макрос код цифры пять, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 6

Six:

PRESS\_KEY 54, 0, 60, 6 ;Передаём в макрос код цифры шесть, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 7

Seven:

PRESS\_KEY 55, 0, 70, 7 ;Передаём в макрос код цифры семь, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 8

Eight:

PRESS\_KEY 56, 0, 80, 8 ;Передаём в макрос код цифры восемь, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма обработки нажатия клавиши 9

Nine:

PRESS\_KEY 57, 0, 90, 9 ;Передаём в макрос код цифры девять, кол-во сотен, десятков и единиц

ret ;Возврат из подпрограммы

;Подпрограмма инициализации дисплея

Init\_lcd:

ldi r25, 0b00000011

rcall Write\_com

ldi r25, 0b00000011

rcall Write\_com

ldi r25, 0b00000011

rcall Write\_com

ldi r25, 0b00000010

rcall Write\_com

ldi r25, 0b00000010

rcall Write\_com

ldi r25, 0b00000001

rcall Write\_com

ldi r25, 0b00000000

rcall Write\_com

ldi r25, 0b00001100

rcall Write\_com

ret ;Возврат из подпрограммы

;Подпрограмма записи данных в регистр данных ЖКИ

Write\_dat:

;Запись старшей тетрады

swar r25 ;Перестановка тетрад

mov r26, r26 ;Копирование во временный рег.

out PORTA, r25 ;Загрузка данных в порт А

sbi PORTA, 5 ;Установка 5 бита в порту А(E=1)

sbi PORTA, 4 ;Установка 4 бита в порту А(RS=1)

DELAY 10 ;Задержка

cbi PORTA, 5 ;Сброс 5 бита в порту А(E=0)

;Запись младшей тетрады

swar r25 ;Перестановка тетрад

mov r26, r25 ;Копирование во временный рег.

out PORTA, r26 ;Загрузка данных в порт А

sbi PORTA, 5 ;Установка 5 бита в порту А(E=1)

sbi PORTA, 4 ;Установка 4 бита в порту А(RS=1)

DELAY 10 ;Задержка

cbi PORTA, 5 ;Сброс 5 бита в порту А (E=0)

ret ;Возврат из подпрограммы

;Подпрограмма записи команды в регистр команд ЖКИ

Write\_com:

```
;Запись старшей тетрады
swap r25      ;Перестановка тетрад
mov r26, r25   ;Копирование во временный рег.
out PORTA, r26 ;Загрузка команды в ПОРТ A
sbi PORTA, 5    ;Установка 5 бита в порту A(E=1)
cbi PORTA, 4    ;Сброс 4 бита в порту A(RS=0)
DELAY 10       ;Задержка
cbi PORTA, 5    ;Сброс 5 бита в порту A(E=0)
```

```
;Запись младшей тетрады
swap r25      ;Перестановка тетрад
mov r26, r25   ;Копирование во временный рег.
out PORTA, r26 ;Загрузка команды в ПОРТ A
sbi PORTA, 5    ;Установка 5 бита в порту A(E=1)
cbi PORTA, 4    ;Сброс 4 бита в порту A(RS=0)
DELAY 10       ;Задержка
cbi PORTA, 5    ;Сброс 5 бита в порту A(E=0)
```

ret ;Возврат из подпрограммы

;Подпрограмма задержки

Long\_delay:

```
ldi Temp, 7
delay_1:
    DELAY 255 ;Задержка
    dec Temp
    brne delay_1
```

ret ;Возврат из подпрограммы

;Подпрограмма вывода 'STELLAZH;'

Stellazh\_info:

```
ldi r25, 83    ;Вывод на экран символа S
rcall Write_dat
ldi r25, 84    ;Вывод на экран символа T
rcall Write_dat
ldi r25, 69    ;Вывод на экран символа E
rcall Write_dat
ldi r25, 76    ;Вывод на экран символа L
rcall Write_dat
ldi r25, 76    ;Вывод на экран символа L
rcall Write_dat
ldi r25, 65    ;Вывод на экран символа A
rcall Write_dat
ldi r25, 90    ;Вывод на экран символа Z
rcall Write_dat
ldi r25, 72    ;Вывод на экран символа H
```

```

    rcall Write_dat
    ldi r25, 58 ;Вывод на экран символа :
    rcall Write_dat
ret ;Возврат из подпрограммы

```

;Подпрограмма вывода 'DOSTAVKA;'

Dostavka\_info:

```

    ldi r25, 68 ;Вывод на экран символа D
    rcall Write_dat
    ldi r25, 79 ;Вывод на экран символа O
    rcall Write_dat
    ldi r25, 83 ;Вывод на экран символа S
    rcall Write_dat
    ldi r25, 84 ;Вывод на экран символа T
    rcall Write_dat
    ldi r25, 65 ;Вывод на экран символа A
    rcall Write_dat
    ldi r25, 86 ;Вывод на экран символа V
    rcall Write_dat
    ldi r25, 75 ;Вывод на экран символа K
    rcall Write_dat
    ldi r25, 65 ;Вывод на экран символа A
    rcall Write_dat
    ldi r25, 58 ;Вывод на экран символа :
    rcall Write_dat
ret ;Возврат из подпрограммы

```

;Подпрограмма вывода 'GRUZ DOSTAVLEN'

Gruz\_info:

```

    ldi r25, 0 ;Вывод на экран символа пробел
    rcall Write_dat
    ldi r25, 71 ;Вывод на экран символа G
    rcall Write_dat
    ldi r25, 82 ;Вывод на экран символа R
    rcall Write_dat
    ldi r25, 85 ;Вывод на экран символа U
    rcall Write_dat
    ldi r25, 90 ;Вывод на экран символа Z
    rcall Write_dat
    ldi r25, 0 ;Вывод на экран символа пробел
    rcall Write_dat
    ldi r25, 68 ;Вывод на экран символа D
    rcall Write_dat
    ldi r25, 79 ;Вывод на экран символа O
    rcall Write_dat
    ldi r25, 83 ;Вывод на экран символа S

```

```
rcall Write_dat  
ldi r25, 84 ;Вывод на экран символа T  
rcall Write_dat  
ldi r25, 65 ;Вывод на экран символа A  
rcall Write_dat  
ldi r25, 86 ;Вывод на экран символа V  
rcall Write_dat  
ldi r25, 76 ;Вывод на экран символа L  
rcall Write_dat  
ldi r25, 69 ;Вывод на экран символа E  
rcall Write_dat  
ldi r25, 78 ;Вывод на экран символа N  
rcall Write_dat  
ret ;Возврат из подпрограммы
```